深層学習を用いたパッシブフィンガープリンティング手法の提案と実装

北條　大和[2,a)]　齋藤　祐太[2]　齋藤　孝道[1]

概要：一部の Web 広告事業者は，ユーザに対して効果的な広告を配信するために，ブラウザフィンガープリントを利用している．ブラウザフィンガープリントの採取手法は大きく分けて，アクティブとパッシブの 2 種類が存在する．1 つ目のアクティブフィンガープリンティングは，JavaScript や CSS を利用してユーザの端末の情報を採取する手法である．2 つ目のパッシブフィンガープリンティングは，HTTP リクエスト時に，Web サーバがブラウザから受け取った HTTP ヘッダなどから情報を採取する手法である．パッシブフィンガープリンティングはアクティブフィンガープリンティングに比べ採取可能な情報が少なく，ブラウザの識別が困難であるとされている．一方で，アクティブフィンガープリンティングに比べ採取時間が短く，JavaScript の利用を拒否するブラウザからも採取が可能という利点が存在する．本論文では，パッシブフィンガープリンティングで採取可能な情報のうち，タイムスタンプや User-Agent 文字列，IP アドレスのみを用いて，深層学習によりモバイル端末の識別を行った．結果として，F1 値が 0.99 以上という精度を得ることができた．

Passive Fingerprinting enforced with Deep Neural Network

Yamato Hojyo[2,a)]　Yuta Saito[2]　Takamichi Saito[1]

Abstract: Some ad companies use browser fingerprinting to provide advertisements to the user effectively. There are two types of methods to collect browser fingerprints: first one is active fingerprinting. The method collects web viewer's fingerprints using with JavaScript or CSS. Second is passive fingerprinting. The method uses only the HTTP header received on a web server at the time of HTTP request. In passive fingerprinting, there can get a little information, and the accuracy of identification is considered to be weak, although collection time is shorter than active fingerprinting. However, passive fingerprinting has a significant advantage, i.e., it can collect fingerprint without JavaScript or its processing time. In this paper, we identified mobile devices with deep learning using only with a timestamp, user agent strings, and IP address in passive Fingerprinting. As a result, we obtained the F1 value of 0.99 or more.

1. はじめに

Web サーバにアクセスしてきたブラウザを識別するブラウザフィンガープリンティングという技術が存在する．この技術は，ブラウザから採取可能な情報を複数利用し，ブラウザごとの情報の組み合わせの差異により，個々のブラウザを識別する技術である．

ブラウザフィンガープリンティングは Web 広告事業者によるターゲティング広告やリスクベース認証に利用されている．Englehardt ら [1] は，2016 年 1 月の調査で，Alexa のランキングトップ 100 万サイトの内，約 1.6%がブラウザフィンガープリンティングを利用していることを明らかにした．また，バージョン 2.0 以降の ITP が採用されている Safari ブラウザでは，サードパーティクッキーの利用は付与された瞬間から即時無効とされ，ターゲティング広告の配信は制限される．その結果クッキーを使用しない識別技術であるブラウザフィンガープリンティングを利用する事業者は，今後増える可能性がある．

[1] 明治大学
Meiji University
[2] 明治大学大学院
Graduate School of Meiji University
[a)] ce195027@meiji.ac.jp

ブラウザフィンガープリンティングには，JavaScript や CSS を用いてブラウザから情報を採取するアクティブフィンガープリンティング，および，ブラウザから送信される HTTP リクエストのヘッダ情報のみを使用するパッシブフィンガープリンティングの 2 種類が存在する．ユーザ側のブラウザで JavaScript の実行を拒否することで，ユーザはアクティブフィンガープリンティングを拒否することができる．一方で，パッシブフィンガープリンティングで採取できる情報は Javascript を用いて採取できる情報と比べて少なく，高精度なブラウザの識別が困難であるとされている．また，パッシブフィンガープリンティングはブラウザが Web サーバに接続した時点で情報の採取が完了するので，短時間で情報を採取できる．加えて，JavaScript の実行を必要としないので，JavaScript の実行を拒否しているブラウザからも情報を採取できる．

本論文では，モバイル端末上のブラウザに限定し，パッシブフィンガープリンティングによる実験を行った．タイムスタンプ，IP アドレス，User-Agent 文字列の情報のみ利用し，深層学習（Deep Neural Network: DNN）で学習した予測モデルで，高精度の識別を試みた．

実験の概要を図 1 に示す．複数のモバイル端末上のブラウザからのアクセスデータ 22,005,939 件を保存し，データセットとする．データセットからランダムに抽出したアクセスデータ 2 件の組み合わせを複数作成することで，深層学習に使用する学習用のデータと，予測モデルの性能評価に使用するテストデータを作成する．深層学習で学習した後，テストデータを用いて，端末（ブラウザ）の識別が可能かを検証する．

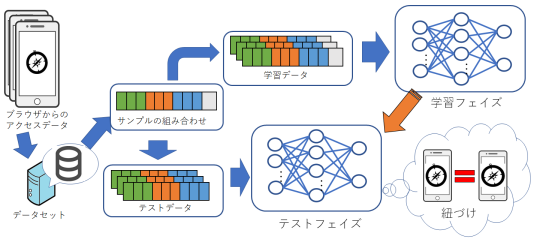

図 1　実験概要

2. ブラウザフィンガープリンティング

ブラウザから採取した情報の組み合わせによって，端末上のブラウザを識別する技術を，ブラウザフィンガープリンティング（以降，フィンガープリンティングと呼ぶ）という．また，フィンガープリンティングのためにブラウザから採取する情報を特徴点と呼び，特徴点の値や，特徴点の組み合わせをブラウザフィンガープリント（以降，フィンガープリントと呼ぶ）という．

Eckersley ら [2] は，フィンガープリントを採取するサイトを構築し，94.2%のフィンガープリントがユニークであることを示した．また，Laperdrix ら [4] は，フランスの Web サイトにおいて，2,067,942 件のフィンガープリントを採取し，大規模なデータにおけるフィンガープリントのユニーク性や，最新の Web 技術におけるフィンガープリンティングの有用性について調べた．結果として，デスクトップやラップトップマシンのフィンガープリントは 35.7%，モバイル端末のフィンガープリントは 18.5%がユニークであった．大規模なデータにおけるフィンガープリントは有用ではなく，フィンガープリンティングによるトラッキングの危険性は低いと論じた．この研究では先行研究と比較するために，特徴点は [2] と同じものを使用しているので，本論文で使用している IP アドレスは，この研究では使用されていない．

フィンガープリンティングはその手法に基づき，2 種類に分類される．2 種類の分類を以下に示す．

(1) アクティブフィンガープリンティング
(2) パッシブフィンガープリンティング

アクティブフィンガープリンティングは，JavaSctipt や CSS を端末上のブラウザで実行させることで採取可能な，画面解像度やフォントリストなどの情報を利用するフィンガープリンティングである．アクティブフィンガープリンティングはユーザがブラウザで JavaScript の実行を拒否することにより，フィンガープリントの採取を拒否することが可能である．また，パッシブフィンガープリントに比べ採取可能な特徴点の種類が多く，ブラウザの識別において有用な情報を多く採取できる．

田邉ら [6] は，フィンガープリンティングにおいて，特徴点の最良の組み合わせを分析した．最良とされた組み合わせで使用される特徴点の多くが，JavaScript から採取可能な特徴点であったことを示している．

パッシブフィンガープリンティングとは，ブラウザから Web サーバへの通信の際に送信される HTTP ヘッダの情報や，IP アドレス，タイムスタンプなどの情報のみを利用するフィンガープリンティングである．その他にも，p0f[5] を利用した手法が知られている．p0f は，主に TCP/IP のパケットを解析して，Web クライアントの OS を推定する手法である．

高橋ら [7] は，JavaScript や CSS を利用せず，HTTP ヘッダのみから採取可能な特徴点のみを利用し，パッシブフィンガープリンティングの実験を行った．結果として，特徴点の中でも特にグローバル IP アドレスと User-Agent 文字列の情報が識別において有用であることを示した．

3. 実験データ

3.1 実験に使用したサンプル

本論文では，複数のモバイル端末のブラウザから，15 日間採取した 22,005,939 件のアクセスデータ（以降，データセットと呼ぶ）を実験に使用した．また，アクセス元を識

別するために，端末ごとに異なる固有の端末識別子を付与した．端末識別子は全部で 616,335 種類だった．

なお，今回の実験の際には，端末識別子において以下の特徴を持つ端末識別子を除き実験を行った．

- 他の端末識別子と比較して，出現回数が飛び抜けて大きいもの
- 端末識別子の出現回数が，他の端末識別子と全く同じ回数のもの
- 全てのタイムスタンプの値が，他の端末識別子と全く同じ値を取るもの

1 つ目の特徴は，スクリプトを用いて自動でアクセスしてきた端末からのアクセスである可能性があると判断し，削除した．2 つ目と 3 つ目の特徴は，別々の端末が全く同じ動作をする可能性は低いと判断し，削除した．

上記の特徴を持つ端末識別子を除き，全部で 596,970 種類の端末識別子を使用する．端末識別子の除去に伴い，データセットのデータ数は全部で 19,494,438 である．端末ごとのアクセス数（端末識別子の出現回数）を集計し，最大アクセス数，最小アクセス数，中央値，平均値について表 1 に示す．

なお，端末識別子は，深層学習を行う際の教師データの作成にのみ使用する．

表 1 個体識別子の出現回数に関する情報

種類	最大アクセス数	最小アクセス数	中央値	平均値
596,970	8856	1	9.0	32.656

表 2 に，データセットのアクセスデータが持つ特徴点を示す．

表 2 データセットのアクセスデータが持つ特徴点

特徴点	例
タイムスタンプ	1488898808
IP アドレス	192.168.100.1
端末識別子	123...abc
OS 名	iOS
OS のバージョン	10.0.2
機種名	SOV32

3.2 特徴点の詳細

本節では，それぞれの特徴点について説明する．

3.2.1 タイムスタンプ

タイムスタンプの期間は連続した 15 日間で，全て unixtime 形式で保存している．日付ごとのアクセス数を調べた結果を，表 3 に示す．

表 3 の通り，15 日間ほぼ均等にアクセスがあったことがわかる．

表 3 日付ごとのアクセス数の割合

日付	アクセス数	全体の割合
1 日目	1,411,274	0.07239
2 日目	1,357,015	0.06961
3 日目	1,342,540	0.06886
4 日目	1,325,110	0.06797
5 日目	1,320,750	0.06775
6 日目	1,315,337	0.06747
7 日目	1,310,509	0.06722
8 日目	1,305,029	0.06694
9 日目	1,296,125	0.06648
10 日目	1,283,573	0.06584
11 日目	1,283,206	0.06582
12 日目	1,252,731	0.06426
13 日目	1,240,186	0.06361
14 日目	1,227,289	0.06295
15 日目	1,223,764	0.06277

3.2.2 IP アドレス

データセット中の IP アドレスは全てグローバル IP アドレスであり，1,158,232 種類存在する．

3.2.3 OS 名・OS バージョン

データセット中の OS は，Android と iOS のみである．それぞれ，iOS のバージョンが 55 種，Android のバージョンが 31 種存在する．表 4 に，データセットの OS の割合と，各 OS ごとに上位 3 つの OS バージョンとその割合を示す．

表 4 OS，OS のバージョンの分布（各 OS ごとに上位 3 つ）

OS の種類	OS の割合	OS バージョン	バージョンの割合
Android	0.405	Android6.0.1	0.128
		Android6.0	0.082
		Android5.0.2	0.055
iOS	0.595	iOS10.2.1	0.335
		iOS10.2	0.065
		iOS10.1.1	0.048

3.2.4 機種名

データセット中に機種名は，3.2.3 節で示した通り，Android と iOS の機種のみ存在する．それぞれ，Android は 930 種，iOS は 16 種存在する．機種名の分布の内，上位 3 種を表 5 に示す．

表 5 機種名の分布（上位 3 種）

OS の種類	機種名	OS 内の割合	全体の割合
Android	SOV32	0.103	0.042
	SOV33	0.042	0.017
	SOV34	0.039	0.016
iOS	iPhone	0.991	0.590
	iPad	0.005	0.003
	iPod	0.003	0.002

3.3 特徴点の生成

表 2 で示した特徴点に加えて，特徴点の値を用いて新たな特徴点を生成した．表 6 に，生成した特徴点を示す．

表 6 生成した特徴点

元の特徴点	生成した特徴点
タイムスタンプ	年, 月, 日, 時, 分, 秒, 曜日
IP アドレス	第 1 オクテット，第 2 オクテット 第 3 オクテット，第 4 オクテット
IP アドレス	ISP 名
IP アドレス	国名，都市名，市区町村名，緯度，経度
OS バージョン	Major，Minor，Maintenance

3.3.1 ISP 名

ISP 名は，pyisp[8] を使用して，IP アドレスから取得した．データセットには，ISP 名は全部で 785 種類存在する．ISP 名に関する集計結果の内，上位 5 つを表 7 に示す．

表 7 ISP 名の分布 (上位 5 つ)

ISP 名	割合
KDDI KDDI CORPORATION, JP	0.3030
GIGAINFRA Softbank BB Corp., JP	0.1791
OCN NTT Communications Corporation, JP	0.1079
DOCOMO NTT DOCOMO, INC., JP	0.1071
JTCL-JP-AS Jupiter Telecommunication Co. Ltd, JP	0.0426

本サンプルの多くが国内からのアクセスであり，ISP 名の出現回数の上位 5 つの内，3 つが日本国内の大手キャリアで占められていることがわかる．

3.3.2 国名や緯度など

国名や緯度などは GeoIP2[9] を使用し，IP アドレスから以下の情報を取得した．GeoIP2 は MaxMind 社の GeoLite2 のデータベースを使用するライブラリであり，GeoLite2 に保存されている位置情報は過去のある時点での情報である．そのため，データセットに含まれる 15 日間全てのフィンガープリントに対して，正確な位置情報を得ている可能性は低い．

- 国名
- 都道府都市名
- 市区町村名
- 緯度および経度

都市名の特徴点の値は，2,252 種存在する．表 8 に，アクセス元の都市の分布を示す．

表 8 アクセス元の都市の分布（上位 5 つのみ）

都市名	アクセス数	割合
東京	5,530,707	0.2837
大阪	1,647,389	0.0845
横浜	745,762	0.0383
埼玉	568,085	0.0291
名古屋	531,816	0.0273

表 8 から，全アクセスの約 28%が東京からのアクセスであるのに対し，それ以外の都市からのアクセスは広く分布していることがわかる．

3.3.3 特徴点の変化について

本節では，15 日間で端末の特徴点が，どの程度変化したかを示す．特に変化しやすいと考えられる 6 種類の特徴点に関して，端末識別子ごとに特徴点の変化回数を計算した．端末識別子ごとの特徴点の変化の詳細を表 9 に記載する．

表 9 において，mean は平均値を表し，std は標準偏差を表す．また，25%と 50%，75%は，それぞれ四分位数を表す．表 9 に，各端末識別子の特徴点がどれほどの種類の値を持っていたのかを示した．例えば値が 2 の場合は，2 種類の特徴点が出現し少なくとも 1 度は特徴点が変化したことを表す．

表 9 から，ISP，OS のバージョン，機種名，User-Agent の中央値は 1.0 である．また標準偏差も小さく，多くの端末は特徴点がほとんど変化しなかったことを表している．

4. 実験について

本節では実験の詳細，およびデータセットのアクセスデータを深層学習のために，ベクトルデータに変換する方法を説明する．

4.1 実験概要

本論文では，深層学習を利用したパッシブフィンガープリンティングによりブラウザの識別が可能か実験した．過去のアクセスデータを深層学習で学習することで，新規のアクセスデータからブラウザをどの程度識別できるのか検証した．また，過去のアクセスデータと新規のアクセスデータを明確に区別するために，15 日間のデータセットをタイムスタンプの値にもとづき前半 7 日間と，後半 8 日間に分割した．

実験では，以下の 3 点について検証した．

- 過去のアクセスデータから，新規のアクセスデータを識別可能か
- 一度目のアクセスから次回アクセスまでの期間と，識別精度の相関があるのか
- パッシブフィンガープリンティングによるブラウザの識別において，有用な特徴点は何か

4.2 ベクトルデータの作成

深層学習のために，アクセスデータをベクトルデータに変換する手順を説明する．

本実験では，任意の 2 件のアクセスデータを組み合わせ，一次元のベクトルデータを作成した．このベクトルデータは組み合わせたそれぞれのアクセスデータのフィンガープリントを比較した情報を示す．

実験に使用する一次元のベクトルデータは以下の手順で

表 9　端末識別子ごとの特徴点の変化回数

	ISP	IP アドレス	OS のバージョン	機種名	User-Agent 文字列	都市
mean	1.4401	4.0608	1.020	1.0019	1.0223	2.3350
std	0.6246	5.1158	0.1446	0.0452	0.1500	1.8321
min	1.0	1.0	1.0	1.0	1.0	1.0
25%	1.0	1.0	1.0	1.0	1.0	1.0
50%	1.0	2.0	1.0	1.0	1.0	2.0
75%	2.0	5.0	1.0	1.0	1.0	3.0
max	13.0	329.0	7.0	6.0	7.0	74.0

作成した．

(1) データセットの中からランダムに 2 件のアクセスデータを抽出し，結合した組を作成

(2) 作成した 2 件のアクセスデータの組における，表 2 と，表 6 の値において，2 件の特徴点を比較した情報を表す特徴点を新たに付加

(3) 2 件のアクセスデータが同一の端末から送信されたアクセスデータかどうかを表す正解ラベルを端末識別子をもとに作成

特徴点の値を比較した情報を表す特徴点の種類を表 10 に示す．

表 10　比較した情報を表す特徴点

比較した特徴点	比較した情報を表す特徴点
IP アドレス	値の一致有無
	オクテットごとの一致の有無
	IP アドレスを連結した文字列
OS 名	値の一致有無
OS バージョン	値の一致有無
	メジャーバージョンの一致有無
	マイナーバージョンの一致有無
	メンテナンスバージョンの一致有無
機種名	値の一致有無
タイムスタンプ	年, 月, 日, 時, 分, 秒, 曜日の一致有無
	日にちと時刻の差
都市名・市区町村名	値の一致有無
	平面上の二点間の直線距離

また，図 2 に，ベクトルデータの作成手順の概念図を示す．

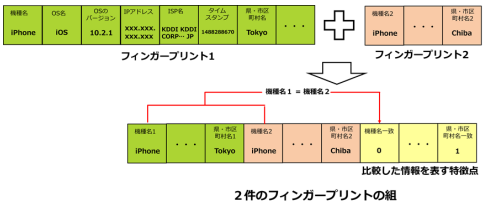

図 2　ランダムに抽出した 2 件のアクセスデータの組の作成

2 件のアクセスデータを結合した組に比較した情報を持つ特徴点を付与した後，深層学習に使用するために，特徴点を数値化した．数値化の詳細を表 11 に示す．

また，図 3 に，作成したベクトルデータを数値化する概念図を示す．

表 11　ベクトルデータの数値化

ハッシュ化し数値化する特徴点	OS 名
	OS バージョン
	機種名
	ISP 名
	国名
	都市名
	市区町村名
	IP アドレスを連結した組
0 or 1 で示す特徴点	値の一致有無を示す特徴点
0 ～ 6 で示す特徴点	曜日
そのまま使用する特徴点	その他の特徴点

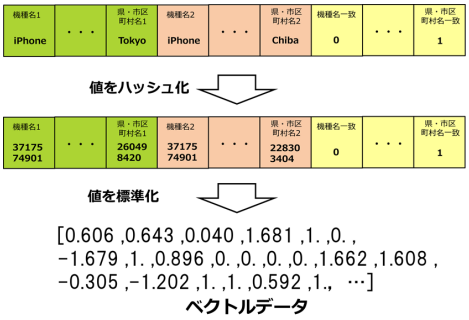

図 3　ベクトルデータの作成

4.2.1　教師データの作成

4.2 節の手順で作成したベクトルデータに対して教師データを作成する方法を説明する．

3.1 節で述べたとおり，データセットのアクセスデータには，端末固有の文字列として，端末識別子が与えられている．教師データは，ベクトルデータにおける 2 件のアクセスデータの端末識別子の一致の有無に応じて作成した．教師データは，組み合わせた 2 件のアクセスデータが同一の端末から送信されたものなのかどうかを表しており，一致していれば正解ラベルとして 1 を，一致していなければ不正解ラベルとして 0 を付与した．

4.3　ニューラルネットワークの構造

実験に使用したニューラルネットワークは，三層の中間層を持ち，各層は全結合層である．損失関数は交差エントロピー関数，最適化関数には Adam を使用した．また，本実験ではデフォルトのパラメータを使用した．ニューラルネットワークの構造について表 12 に示す．

表 12　ニューラルネットワークの構造

層	ユニット数	活性化関数	DropOut
入力層	特徴点の数	relu	無し
中間層 1	1024	relu	0.3
中間層 2	2048	relu	0.2
中間層 3	1024	relu	0.2
出力層	2	sigmoid function	無し

4.4　予測モデルの作成

実験のための予測モデルの作成および学習の手順について説明する．

深層学習は，Android のみの場合，iOS のみの場合，両方の OS を使用した場合の 3 パターン行った．3 パターンそれぞれ，教師データは正解と不正解のラベルの数が均等になるように，各 2,000 万のベクトルデータを作成し，これらを学習データとした．過去のアクセスデータを学習し，新規のアクセスデータからブラウザを識別することが実験の目的なので，学習データは全て，前半 7 日間のアクセスデータからランダムに 2 件ずつ抽出し，組み合わせた．

作成したベクトルデータを用いて，教師あり学習の深層学習により 3 パターンの予測モデルを作成した．

4.5　実験の詳細

本論文で行う 3 つの実験の詳細を説明する．

4.5.1　実験 1

実験 1 では，過去のアクセスデータを学習することで，新規のアクセスデータが与えられたときに，ブラウザをどの程度識別できるのか検証した．学習データとは別に，iOS のみの場合，Android のみの場合，両方の OS を使用した場合の 3 パターンに対して各 2,000 万ずつベクトルデータを作成し，これらをテストデータとした．テストデータは全て，前半 7 日間のアクセスデータと後半 8 日間のアクセスデータからランダムに 1 件ずつ組み合わせ，教師データの正確と不正解のラベルが均等になるように作成した．節 4.4 で学習した各 OS ごとのモデルを使用し，各 2,000 万個ずつのテストデータで精度を検証した．

4.5.2　実験 2

実験 2 では，フィンガープリントを採取した日から，次回アクセスまでの間隔と識別精度に相関があるかを調べた．テストデータとして組み合わせた 2 件のアクセスデータのタイムスタンプの差を取ることで，日付の差を計算しアクセス間隔の値とする．実験 1 と同様の手順で，実験 2 のテストデータは実験 1 で使用したテストデータとは別に作成し，アクセス間隔が 1 日から 14 日まで開いた場合の計 14 グループに分類した．各グループ 200 万個ずつ，教師データのラベルが均等になるようにテストデータを各 OS ごとに 3 パターン作成した．節 4.4 で学習した各 OS ごとのモデルを使用して，各 OS ごとに 200 万ずつ 14 グループのテストデータで精度を検証した．

4.5.3　実験 3

実験 3 では，ランダムフォレストを利用し，パッシブフィンガープリンティングにおいて有用な特徴点を調べた．実験 1 で使用した，iOS のみの場合，Android のみの場合，両方の OS を利用した場合の 3 パターンで使用したテストデータをランダムフォレストの学習データとして使用した．scikit-learn の RandomForestClassifier を使用して決定木探索を行い，各 OS の 3 パターンを対象として，パッシブフィンガープリンティングの識別において重要な特徴点を調べた．

5.　実験結果

5.1　識別精度算出に使った指標

モデルの予測値と教師データに基づき，Precision，Recall，Accuracy，F_1 値をそれぞれ算出し，使用する．以下に，*Precision*，*Recall*，*Accuracy*，F_1 値を求める式を示す．

$$Precision = \frac{|TP|}{|TP + FP|}$$

$$Recall = \frac{|TP|}{|TP + FP|}$$

$$Accuracy = \frac{|TP + TN|}{|TP + FP + FN + TN|}$$

$$\mathrm{F}_1 = \frac{2 * Precision * Recall}{Precision * Recall}$$

上記の識別精度の算出の際には，表 13 に基づき，TP や TN を算出する．作成した予測モデルの予測値は，フィンガープリントによる判定結果，教師データはデータセットの端末それぞれに付与した端末識別子の一致有無とする．なお，ユーザ側で意図的にフィンガープリントを変更（偽造）した場合は考慮しない．

表 13　TP，TN，FP，FN の分類

		端末識別子	
		同一	異なる
フィンガープリントによる判定	同一	TP	FP
	異なる	FN	TN

5.2　実験 1 について

実験 1 では，深層学習を利用し，パッシブフィンガープリンティングの実験を行った．実験 1 の結果を表 14 に示す．

表 14　実験 1 の結果

各評価指標における数値	両 OS	Android	iOS
Precision	0.992	0.999	0.999
Recall	0.996	0.999	0.994
Accuracy	0.994	0.999	0.997
F_1	0.994	0.999	0.997

表 14 により，iOS のみの場合，Android のみの場合，両

方の OS を利用した場合，全てにおいて，0.99 以上と良好であった．

この結果から，タイムスタンプ，User-Agent 文字列，IP アドレスのみを用いたパッシブフィンガープリンティングでも，深層学習を用いたブラウザの識別が可能であると言える．

5.3 実験 2 について

実験 2 では，前回のアクセスから次回アクセスまでの間隔と識別精度の相関を調べた．表 15 に，実験結果を示す．

表 15 実験 2 の結果（数値は全て F_1 を使用する）

期間	両 OS	Android	iOS
1 日	0.975	0.964	0.973
2 日	0.978	0.964	0.899
3 日	0.985	0.938	0.893
4 日	0.984	0.963	0.897
5 日	0.978	0.930	0.821
6 日	0.982	0.958	0.843
7 日	0.959	0.970	0.775
8 日	0.958	0.965	0.808
9 日	0.918	0.938	0.770
10 日	0.914	0.959	0.727
11 日	0.930	0.959	0.725
12 日	0.975	0.930	0.685
13 日	0.947	0.940	0.794
14 日	0.961	0.965	0.855

表 15 から，iOS の場合，次回アクセスまでの期間が長くなるにつれ，Android と比較して，精度が低下する傾向がある事がわかった．

5.4 実験 3

実験 3 では，4.5.1 に示した，3 パターンの識別精度の検証において，ブラウザの識別に有用な特徴点を調べた．今回，ランダムフォレストを用いた．重要な特徴点とされた特徴点を上位 10 位まで表 16 に示す．

表 16 の 3 パターンどれについても，OS バージョンの一致有無が，上位 2 位以内に入っており，特に有用な特徴点だということがわかった．

6. 考察

本節では，高精度の識別が可能であった原因を考察する．

6.1 実験 1 について

実験 1 では，4.5.1 に示した，3 パターンの精度を検証した．その結果，識別精度において全て 0.99 以上と良好であった．この原因について考察する．

3 節でのデータの分析から，以下のことがわかった．

(1) 3.3.2 節に示したアクセス元の都市を見ると，2,252 種の都市の中で，東京からのアクセスが 28%であったが，その他のアクセスは全て 1%を下回っていた．

(2) 3.3.3 節で示した，特徴点の値の変化を見ると，IP アドレスに比べて User-Agent 文字列や ISP の変化回数は少なかった．

(3) 3.2.4 節の機種名の特徴点と，3.2.3 節の OS のバージョンをメンテナンスバージョンまで見た際に，特徴点の値の組み合わせが多かった．

以上のことの組み合わせにより，高精度で識別できるほどの多様性がアクセスごとに存在したことが推察される．

実験 1 では，先行研究 [7] と違い，HTTP ヘッダから採取可能な情報を利用することなく，タイムスタンプ，IP アドレス，User-Agent 文字列のみを利用した．このことから，タイムスタンプ，IP アドレス，User-Agent 文字列のみでも，十分な情報を持つことがわかる．

6.2 実験 2 について

実験 2 において，iOS のみの識別に限定した際に，アクセス間隔が空くと精度が低下する原因を考察する．

表 4 に関して，1 番多かった OS バージョンの割合を OS ごとに見てみると，Android6.0.1 は約 32%，iOS10.2.1 は iOS 内で約 56%という割合を示していた．Android は，変化しにくい特徴点の値の分散が，iOS に比べて多かったので，長期間の識別しやすかったと考えられる．一方で，3 節で示した通り，iOS は Android に比べて，各特徴点の値の分散が小さい．IP アドレスが変化しやすいことを考えると，その他の機種名や OS のバージョンなどの特徴点の値の分散が少ないことが，長期間の識別ができなかったことの原因だと考えられる．

6.3 実験 3 について

実験 3 の結果を見てみると，Android では機種名が識別に影響のある特徴点とされている一方で，iOS に注目してみると，機種名の特徴点は 10 位以内に入っていない．この結果は，iOS の機種名の 99%が iPhone であるので，識別には有用でなかったためである．

また，表 9 で示した変化しにくい特徴点の多くが，ランダムフォレストにおいて重要な特徴点とされていることがわかる．加えて，変化しやすいと考えられる IP アドレスに関して，第 1 オクテットの値が 3 パターン全てにおいて上位 3 位以内に入っている．IP アドレスの第 1 オクテットから，第 4 オクテットまで値の変化を調べたところ，表 17 のようになった．

表 17 から，IP アドレスの中でも，第 1 オクテットは変化しにくいが，それ以外は変化しやすいことがわかる．また，第 1 オクテットの値は 4,755 種存在し，1,922 種がユニークな値であった．

以上のことから，特徴点の値の種類が多く，値が変化し

表 16　識別において重要な特徴点（上位 10 個）

rank	両方の OS	iOS における重要な特徴点	Android における重要な特徴点
1	OS バージョンの一致有無	IP アドレスの第 1 オクテットの一致有無	機種名の一致有無
2	OS のメンテナンスバージョンの一致有無	OS バージョンの一致有無	OS バージョンの一致有無
3	IP アドレスの第 1 オクテットの一致有無	OS のメンテナンスバージョンの一致有無	IP アドレスの第 1 オクテットの一致有無
4	IP アドレスの一致有無	ISP 名の一致有無	OS のメジャーバージョンの一致有無
5	IP アドレスの第 2 オクテットの一致有無	IP アドレスの一致有無	OS のメンテナンスバージョンの一致有無
6	IP アドレスの第 3 オクテットの一致有無	直線距離	ISP 名の一致有無
7	ISP 名の一致有無	IP アドレスの第 4 オクテットの一致有無	直線距離
8	都市名の一致有無	都市名の一致有無	OS のマイナーバージョンの一致有無
9	OS の一致有無	OS のマイナーバージョンの一致有無	IP アドレスの一致有無
10	OS のマイナーバージョンの一致有無	前半 7 日間サンプルの OS バージョンの値	IP アドレスの第 4 オクテットの一致有無

表 17　IP アドレスのオクテットごとの特徴点の変化

	第 1	第 2	第 3	第 4
mean	1.7632	4.1015	5.5654	6.3958
std	0.9567	3.2787	6.1042	8.2509
min	1.0	1.0	1.0	1.0
25%	1.0	2.0	2.0	2.0
50%	2.0	3.0	3.0	4.0
75%	2.0	6.0	7.0	8.0
max	57.0	89.0	248.0	385

にくい特徴点は，パッシブフィンガープリンティングの識別において，有用な特徴点となる可能性がある．

6.4　サンプルの採取期間と OS のバージョンについて

実験 3 によって，OS のバージョンや IP アドレスに関する特徴点が識別において重要な特徴点ということがわかった．15 日間という短い期間での実験だったことと，表 9 の端末識別子ごとの特徴点の変化にも示した通り，OS のバージョンアップはほとんど無かった．しかし，15 日以上の長期間の識別の際には，OS のバージョンアップなどが起こる可能性は高いと考えられるので，本実験の精度とは異なる結果になる可能性がある．

7.　研究倫理

我々は，Menlo report[3] の精神に則り，倫理的配慮をして実験を行った．実験を行う際，個人識別はせず，プライバシーを遵守した．本論文で使用されたデータセットの提供元はデータセットの利用目的を理解している．また，研究に使用されたデータセットは，学術的な目的にのみ使用し，我々の研究室にて厳重に保管されており，他者への売却および提供をしない．

8.　まとめ

本論文では，タイムスタンプ，IP アドレス，User Agent 文字列から利用できる情報のみを深層学習で学習し，パッシブフィンガープリンティングの実験を行った．結果として，15 日間という期間では Precision，Recall，Accuracy，F_1 値が 0.99 以上という精度でモバイル端末の識別が可能であった．

また，ランダムフォレストを用いて，パッシブフィンガープリンティングの識別において影響のある特徴点を調べた．結果として，パッシブフィンガープリンティングでのモバイル端末の識別において，15 日間という短期間の識別であれば OS のバージョン情報に関する特徴点や IP アドレスに関する特徴点が，識別において影響のある特徴点であることがわかった．

参考文献

[1] S. Englehardt and A. Narayanan. Online tracking: A 1-million-site measurement and analysis. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS ’16, pages 13881401, New York, NY, USA, 2016. ACM.

[2] P. Eckersley, ”How Unique Is Your Web Browser?”, in-Proc. of the 10th international conference on Privacy enhancing technologies (PETS’10), 2010.

[3] Dittrich, D. and Kenneally, E. The Menlo Report: Ethical Principles Guiding Information and Communication Technology Research. U.S. Department of Homeland Security, Aug 2012.

[4] Gómez-Boix, Alejandro and Laperdrix, Pierre and Baudry, Benoit, Hiding in the Crowd: an Analysis of the Effectiveness of Browser Fingerprinting at Large Scale. WWW2018 - TheWebConf 2018 : 27th International World Wide Web Conference, Lyon, France, Apr 2018.

[5] ”Passive OS Fingerprinting: Details and Techniques” http://www.ouah.org/incosfingerp.htm

[6] 田邉一寿, 高橋和司, 安田昂樹, 種岡優幸, 細谷竜平, 小芝力太, 齋藤祐太, 齋藤孝道, ”Browser Finger- printing における特徴の組み合わせに関する考察”, コンピュータセキュリティシンポジウム 2017 論文集 CD-ROM pp.1090-1097, 2017

[7] 高橋和司, 安田昂樹, 種岡優幸, 田邉一寿, 細谷竜平, 野田隆文, 齋藤祐太, 小芝力太, 齋藤孝道，HTTP ヘッダのみを用いた Browser Fingerprinting の考察, 暗号と情報セキュリティシンポジウム (SCIS) 2018 (2018).

[8] pyisp, https://github.com/ActivisionGameScience/pyisp/

[9] GepIP2-python, https://github.com/maxmind/GeoIP2-python/