

深層学習によるアプリログからの特徴抽出の検討

落合 桂一^{1,a)} 山本 直樹¹ 濱谷 尚志¹ 深澤 佑介¹

概要：スマートフォンの利用状況から利用者の状態推定を行う研究が盛んに行われている。従来研究ではカテゴリごとのアプリ利用回数や利用時間などの基本統計量が特徴量として利用されていた。しかしながら、アプリの使用履歴からは基本統計量に限らず、アプリの利用順序やアプリ間の関係性など多様な特徴表現が生成可能であると考えられる。そこで本研究では、深層学習によるアプリ利用ログからの特徴抽出を行い、分類問題により特徴抽出の有効性を評価する。深層学習モデルとして LSTM や Graph Convolutional Networks など複数のモデルを検証する。

1. はじめに

スマートフォンの普及を背景に、スマートフォンの利用状況やスマートフォンに搭載されたセンサにより得られたデータから利用者の状態推定を行う研究が盛んに行われている。例えば、ストレス状態 [1], [2] や注意力 [3], [4], 倦怠度 (Boredom) [5] などユーザの内部状態の推定において、スマートフォンアプリの使用履歴が特徴量として利用されている。表 1 に Android OS のアプリ使用履歴の例を示す。timestamp はアプリを使用した時刻を示し、class name はアプリごとの画面の識別子を示す。このログの例は、ユーザがホーム画面 (com.nttdocomo.android.dhome.HomeActivity) から電話を起動し (com.android.dialer.DialtactsActivity)、通話を行い (com.android.incallui.InCallActivity)、ホーム画面に戻ったことを示す。このようなアプリ使用履歴から、アプリ利用回数や利用時間などの基本統計量を特徴量として状態推定を行う。例えば、Yamamoto らの研究 [2] では、所定時間内における SMS や電話などのコミュニケーションに利用するアプリの起動回数や、業務用アプリケーションの利用回数を特徴量としている。また、Pielot ら [5] は倦怠度をカテゴリ別のアプリ起動回数などを特徴量として推定している。

これらの従来研究では、アプリ使用履歴からの特徴抽出方法として、カテゴリごとのアプリ利用回数や利用時間などの基本統計量が特徴量として利用されていた。また、これらの特徴量は推定対象に応じて人手で設計する必要がある。一方、アプリの使用履歴からは基本統計量に限らず、

アプリの利用順序やアプリ間の関係性など多様な特徴表現が生成可能であると考えられる。ここで、一例としてアプリの利用順序の特徴量化について考える。アプリの利用は表 1 における class name として表現されるため、class name を自然言語処理における単語とみなすと、画面遷移を class name の N-gram として表現することで利用順序を考慮した特徴量化ができると考えられる。アプリの起動回数を集計することは、uni-gram 特徴量を生成していることに相当する。しかしながら、N-gram を特徴量とした場合、アプリは数百万件存在するため、データスパースネス問題が課題となる [6]。

そこで本研究では、密な特徴ベクトルを生成できる深層学習による特徴抽出を提案し有効性を検証する。本研究では複数の深層学習手法を適用し、有効性の比較を行う。1 つ目の手法は、アプリの使用は時系列であることに着目し、時系列データに適した深層学習の手法である Long Short-Term Memory (LSTM) [7] を検討する。2 つ目の手法として、アプリの使用履歴は class name をノードとするとグラフ構造で表現できることに着目し、Graph Convolutional Network (GCN) [8] を検討する。本研究では GCN の一種である Graph Isomorphism Network (GIN) [9] を利用する。これらの深層学習による特徴抽出の有効性を評価するため、Yamamoto らの研究 [2] と同様の分類問題に対して分類精度を比較することで特徴抽出の性能を評価する。

本研究の貢献は以下の通りである。

- ユーザ状態推定問題においてアプリ使用履歴の特徴抽出に深層学習を利用することを提案した。深層学習の手法として LSTM や GCN など複数手法を検討した。
- 時系列データであるアプリ使用履歴に対して GCN を

¹ 株式会社 NTT ドコモ, NTT DOCOMO, INC.

^{a)} ochiaiike@nttdocomo.com

表 1 アプリ使用履歴の例
Table 1 Example of app usage

timestamp	class name
2016-02-17 17:14:44	com.nttdocomo.android.dhome.HomeActivity
2016-02-17 17:14:45	com.android.dialer.DialtactsActivity
2016-02-17 17:14:49	com.android.incallui.InCallActivity
2016-02-17 17:25:44	com.nttdocomo.android.dhome.HomeActivity

適用するため、各 class name をノード、連続して使用された 2 つの class name に対してエッジを張ることでグラフを構築し、ユーザ状態推定をグラフ分類問題として問題定義した。

- 既存のユーザ状態推定問題に深層学習によるアプリ使用履歴の特徴量を加えることで分類精度が向上することを確認した。

本稿の構成は以下の通りである。次章にてユーザ状態推定の既存研究および深層学習による特徴抽出について概観する。次に、3 章で提案手法の詳細を説明し、4 章では実データを用いた評価を行い、既存手法と提案手法の精度について検証する。最後に 5 章にて本研究のまとめおよび今後の課題について述べる。

2. 関連研究

2.1 ユーザ状態推定

アプリ使用ログを用いたユーザ状態推定の研究では、推定対象としてデモグラフィック属性 [10] や性格 [11] などの静的な状態や、ストレス状態 [1], [2], 注意力 [3], [4], 倦怠度 [5] など動的な状態推定が行われている。これらの研究では、アプリの利用回数や利用時間などの基本統計量の特徴量としている研究が多い。深層学習を用いてアプリ使用ログから特徴抽出を行なった研究はない。

2.2 深層学習による特徴抽出

Multi-Layer Perceptron (MLP) や Convolutional Neural Network (CNN) など深層学習モデルは、入力データの特徴抽出を行う部分と、分類器としての機能を有する部分で構成される。深層学習は特徴表現学習とも呼ばれ、推定対象に適した特徴抽出器を人手で設計することなく大量のデータから学習できることが特徴である [12]。例えば、画像認識において事前学習した CNN を特徴抽出器として利用したり [13], [14], 自然言語処理において単語や文章から特徴ベクトルを生成するために LSTM モデルが特徴抽出器として利用されたりしている [15]。また、近年グラフ構造に対する深層学習モデルに関する研究も盛んに行われ、特にグラフやノードの特徴表現を得るために畳み込みを利用する GCN が注目されている [8]。

これらの手法は、CNN や LSTM は画像認識や自然言語処理の分野で主に研究が行われており、また、GCN に関

する研究では論文の引用関係やタンパク質の構造など特徴量量化して検証が行われているが、アプリ使用履歴に対する有効性は明らかではない。

3. 提案手法

本研究では、深層学習の手法として LSTM および GCN によるアプリ使用履歴の特徴抽出を検討する。本章では、本研究で対象とする問題設定、および各手法の詳細を説明する。

3.1 用語定義および問題定義

本節では、用語定義と問題設定について説明する。
定義 1 (アプリ使用ログ) 各アプリ使用ログ l は、ユーザ $u \in U$, timestamp t および画面の識別子である class name c から構成され、 $l = (u, t, c)$ とする。

定義 2 (アプリ使用ログ系列) 特定の時間範囲 (例えば直近 T 時間) のアプリ使用ログ l を timestamp t で時系列にソートしたデータをアプリ使用ログ系列と定義する。すなわち、アプリ使用ログ系列 x は $x_i = \{l_1, l_2, \dots, l_s\}$ とする。ここで i はアプリ使用ログ系列のインデックス、 s は特定の時間範囲におけるアプリ使用ログ数を示す。

(問題定義) アプリ使用ログ系列 x_i が与えられたとき、アプリ使用ログ系列の特徴を表す特徴ベクトル r_i を求める関数 f を求めることが特徴抽出に該当する。特徴ベクトル r_i を入力として分類問題を解いた際に、もっとも分類性能が高くなる関数 f を求めることが本研究で主題とする特徴抽出である。本研究において、関数 f は LSTM (3.2 節) または GCN (3.3 節) で構成する。

3.2 LSTM による特徴抽出

図 1 に LSTM による特徴抽出の概要を示す。アプリ使用ログ系列 x が与えられると、各 class name $l_1.c, l_2.c, \dots$ が時系列順に LSTM モデルに入力され、式 (1) により隠れ層のベクトル h に変換される。

$$h_t = LSTM(h_{t-1}, l_{t,c}) \quad (1)$$

表記は文献 [16] を参考とした。 $t = 0$ から再帰的に上式に従って隠れ層ベクトルが生成される。学習時は図 1 のよう

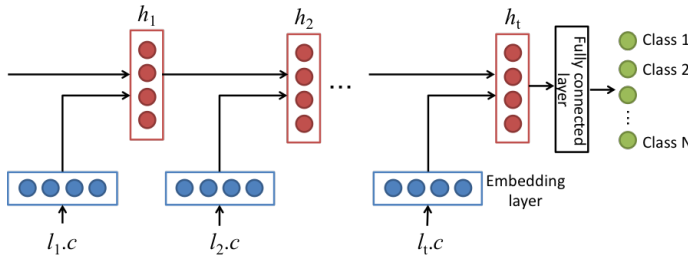


図 1 LSTM による特徴抽出

Fig. 1 Feature Extraction using LSTM

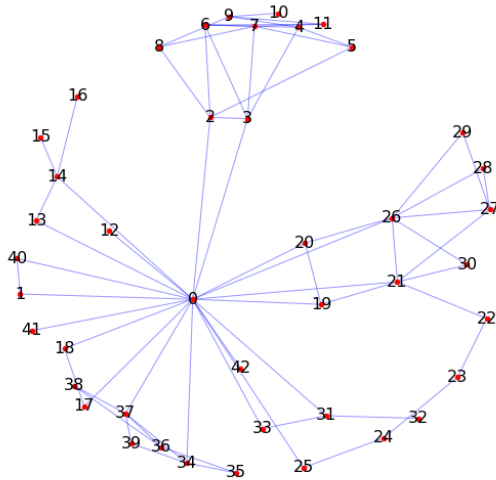


図 2 アプリ使用ログ系列をグラフとして表現した例

Fig. 2 Example of Graph Representation of App Usage

にクラス分類問題として教師あり学習を行う。図 1 では N クラス分類の例を示している。Loss 関数には cross entropy を利用する。特徴抽出器として利用する場合は、全結合層の 1 つ前の層で出力される隠れ層ベクトル h_t を一連のアプリ使用履歴に対する特徴量として抽出する。すなわち、入力 x から特徴ベクトル $r (= h_t)$ を生成する処理が関数 f に相当し、特徴抽出は以下の式となる。

$$f(x) = LSTM(h_{t-1}, l_{t,c}) \quad (t = 0, \dots, s) \quad (2)$$

3.3 GCN による特徴抽出

本研究では、グラフ構造に対する深層学習手法である GCN[8] による特徴抽出を検討する。GCN とは、グラフにおける隣接ノードの分散表現を段階的に集約することでノードやグラフの分散表現を得る手法である。Algorithm1 に各ノードの分散表現を得る処理の流れを示す。ここで、 $G(V, E)$ はグラフ、 x_v は各ノードの特徴量、 h_v^k は k 層目のノード v に対する特徴量、 $N(v)$ はノード v の隣接ノードを示す。Algorithm1 の 4 行目の処理で周辺のノードを集約する $AGGREGATE_k$ により、隣接ノード $N(v)$ の分散表現を集約し a_v^k を得る。 $AGGREGATE_k$ の処理は、要素ごとの平均や最大、総和などで計算する。次に、5 行目

Algorithm 1 Node embedding generation algorithm [8]

Input: Graph $G(V, E)$, input features $\{x_v, \forall v \in V\}$

Output: Vector representations h_v^K for all $v \in V$

```

1:  $h_v^0 \leftarrow x_v$ 
2: for  $k = 1$  to  $K$  do
3:   for  $v \in V$  do
4:      $a_v^k \leftarrow AGGREGATE_k(\{h_u^{k-1}, u \in N(v)\})$ 
5:      $h_v^k \leftarrow COMBINE_k(h_v^{k-1}, a_v^k)$ 
6:   end for
7:  $h_v^K \leftarrow NORMALIZE(h_v^K), \forall v \in V$ 
8: end for

```

の $COMBINE_k$ では対象ノードの分散表現 h_v^{k-1} と、隣接ノードを集約して得られた分散表現 a_v^k を統合する。 k 層の GCN では k ホップ先のノードの特徴量を集約する。GCN においては、 $AGGREGATE_k$ および $COMBINE_k$ の選択が GCN の性能に重要である。グラフ全体の特徴ベクトルを得るには、この処理の後に全ノードに対する集約処理がさらに必要となる。

$$h_G = READOUT(\{h_v^K | v \in G\}) \quad (3)$$

本研究では、GCN の一種である Graph Isomorphism Network (GIN) [9] を利用する。GIN では $AGGREGATE_k$ および $COMBINE_k$ (Algorithm1 の 4 行目、5 行目) を以下の式で計算する。

$$h_v^k = MLP^k\left((1 + \epsilon^k) \cdot h_v^{k-1} + \sum_{u \in N(v)} h_u^{k-1}\right) \quad (4)$$

ここで MLP は Multi-Layer Perceptron を、 ϵ は学習可能なスカラーを示す。MLP には学習可能な重み行列 W^k が含まれる。また、グラフ全体の特徴ベクトルを得るための $READOUT$ には畳み込みの各層で得られる h_v^k を連結する関数を利用する。すなわち、 $k = 0, 1, \dots, K$ として、入力 x から h_G を生成する式 (5) が特徴抽出に該当する。

$$f(x) = CONCAT\left(READOUT(\{h_v^k | v \in G\})\right) \quad (5)$$

アプリ使用ログ系列 x が与えられたとき、各 class name をノードとし、連続する 2 つの class name に対してエッジを設定することでグラフを構築する。ここで構築されるグラフは無向グラフとなる。図 2 に実際にアプリ使用ログ系列からグラフを構築した例を示す。ノードの番号は各 class name に対して割り当てた ID を示す。図 2 においては、ID 0 がホーム画面を示しており、ホーム画面は各アプリを起動する際に起点となっていることがわかる。このグラフに対して、Algorithm1 の処理を行うことでグラフの特徴ベクトル r が生成される。

3.4 分類器の学習

ここまで説明した LSTM または GIN により、アプリ使用ログ系列 x_i が与えられたとき特徴ベクトル r_i が得ら

れているとする．各アプリ使用ログ系列 x_i に対して，分類問題における教師ラベル y_i が与えられているとすると， $D = \{r_i, y_i\}_{i=1}^N$ を利用して教師あり機械学習モデルを訓練する．対象とする問題によっては，アプリ使用ログ以外のデータ（例えば，後述のストレス状態推定においては加速度センサで得られる特徴量）も利用できる．その場合は，アプリ使用ログ以外のデータから生成された特徴ベクトルと r_i を組み合わせた特徴ベクトルを \hat{r}_i とすると， $D = \{\hat{r}_i, y_i\}_{i=1}^N$ を利用して教師あり機械学習モデルを訓練する．

4. 評価

本章では提案手法の有効性を確認するために行った評価について説明する．本研究では，アプリ使用ログを特徴量として用いているユーザ状態推定のタスクにおいて，深層学習による特徴抽出を行った場合と従来手法で推定精度を比較することで有効性を検証する．

4.1 評価タスク

本研究では Yamamoto ら [2] のストレス状態推定において，アプリ使用ログの基本統計量を特徴量とした場合をベースラインとして，深層学習を用いた手法との推定精度の比較を行う．Yamamoto らの研究の問題設定および利用したデータについて説明する．株式会社 NTT ドコモの R&D 部門の社員 39 名から同意を得て，心拍データおよびスマートフォンで得られる端末利用ログ，センサ，位置情報などを取得した．39 名の研究参加者より，1349 日分のスマートフォンログと 724 日分の心拍データを収集した．ストレス状態は，心拍計で計測した RR 間隔 (RRI: RR-interval) から計算される LF/HF を利用している．研究参加者の所定期間の LF/HF の平均値を計算し，平均以上であれば高ストレス，平均未満であれば低ストレスとラベル付けしている．スマートフォンで得られるログには，加速度や照度などのセンサ，アプリ使用ログ，位置情報，バッテリー残量やイヤホンなど端末状態などがある．これらのデータから 128 次元の特徴量を生成しストレス状態の推定を行っている．アプリ使用ログについては，SMS，電話，業務用アプリ，その他のアプリの起動回数を特徴量としている．起動回数は 1 日を 6 時間ごとに 4 分割 (1Q: 0 時～6 時，2Q: 6 時～12 時，3Q: 12 時～18 時，4Q: 18 時～24 時) した上で，各時間帯で集計した値を利用する．アプリ使用ログを元にした特徴量は 16 次元となる．詳細については，文献 [2] を参照されたい．本研究の評価では，上記の特徴量に加え，アプリ使用ログから深層学習により抽出した特徴量を加えストレス状態を推定した際に，分類精度が向上するか検証を行う．本研究では，アプリ使用ログが存在する日のデータを対象とし，28 名，457 日分のデータを利用した．

4.2 評価条件

LSTM モデルおよび GCN モデルの両方で，後段の分類器にはロジスティック回帰モデルを利用した．Yamamoto らの研究では，128 次元の特徴量と教師データで分類器を学習することに加え，クラスタリングや特徴選択により推定精度の向上を図っていたが，本研究では特徴量追加の効果を見るため，これらの手法は適用せず分類器の学習を行った．評価指標には精度 (accuracy) を用いる．推定精度は 10 分割交差検定により評価した．

LSTM モデルでは，各日のアプリ使用ログの最初から 500 件を利用し，隠れ層の次元数は 16, 32, 64 の 3 パターンで比較を行った．重みの学習には確率的勾配降下法 [17] を利用し，ストレス状態を教師データとして学習を行った．学習エポック数は 10 とした．その後，アプリ使用ログを LSTM モデルに入力し，最終層で得られる隠れ層のベクトルを抽出し，Yamamoto らの手法で利用されている 128 次元の特徴量を連結した特徴量を用いて分類器を学習した．例えば，隠れ層の次元数を 16 とした場合は，16+128=144 次元となる．LSTM モデルの学習では，10 分割交差検定のために 10 グループに分けたデータのうち，9 グループを学習に利用するが，この際，9 グループのデータの 90% を学習データ (training data)，10% を検証用データ (validation data) とした．

GCN モデルでは，はじめに 1 日分のアプリ使用ログから連続する class name に対してエッジを張りグラフを構築した．次に，1 日分のアプリ使用ログを入力にストレス状態を教師データとして学習を行った．式 (4) の MLP は 2 層，畳み込みの層数 (K) は 3 とした．重みの学習には LSTM と同様に確率的勾配降下法を用い，学習エポック数は 10 とした．その後，アプリ使用ログを GCN モデルに入力し，得られた特徴ベクトルを抽出したうえで，Yamamoto らの手法で利用されている 128 次元の特徴量を連結し分類器を学習した．特徴量の次元数は 16, 32, 64 の 3 種類で比較を行った．

4.3 評価結果

図 3 および図 4 に各モデルのロス関数の学習曲線の例を示す．どちらも特徴量の次元数を 16 としたときで，10 分割交差検定のうちの 2 例である．LSTM モデルでは単調減少とはなっていない，一方，GCN モデルでは単調減少となっており，GCN モデルの方が学習が効率的に行えていると考えられる．

表 2 に各手法の推定精度を示す．Yamamoto ら [2] の手法は，128 次元の特徴量と教師データでロジスティック回帰モデルを学習した結果を示す．この手法を以下ベースラインと呼ぶ．LSTM および GCN では，特徴抽出した次元数を変え，従来手法で利用されていた 128 次元の特徴量と連結しロジスティック回帰モデルを学習した結果を示す．

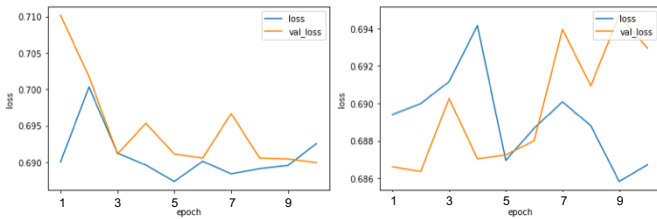


図 3 LSTM モデルの学習曲線

Fig. 3 Learning Curve of LSTM model

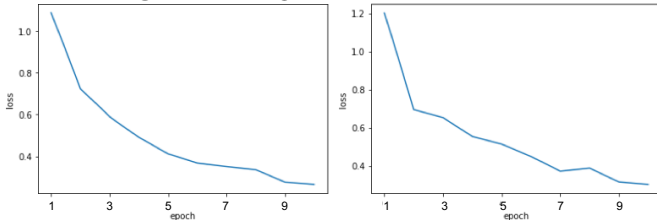


図 4 GCN モデルの学習曲線

Fig. 4 Learning Curve of GCN model

表 2 推定精度の比較

Table 2 Performance Comparison

手法	追加した特徴次元数	精度
Yamamoto ら [2]	-	0.531
LSTM	16	0.495
	32	0.493
	64	0.495
GCN	16	0.542
	32	0.545
	64	0.523

表 2 より, LSTM により特徴抽出を行った場合はすべてのパターンで精度が低下していた。学習曲線から適切に学習が行えていないと考えられるため, その結果として精度が低下したと考えられる。一方, GCN によりアプリ使用ログの特徴量を生成した場合は, 64 次元とした場合以外ではベースライン手法より分類精度が向上していた。そこで, 特徴抽出を行う次元数ごとに生成されるベクトルの内容を確認した。64 次元で生成した特徴ベクトルの値を確認したところ, すべてのサンプルで値が 0 となっている次元が存在していた。一方, 16 次元の特徴ベクトルでは, すべてのサンプルでほぼすべての次元に 0 ではない値が設定されていた。このことから, 冗長な次元が含まれることが精度低下の原因と考えられる。

5. おわりに

本研究では, 深層学習によるアプリ使用ログからの特徴抽出を提案し有効性を検証した。深層学習には, 1 つ目の手法としてアプリの使用は時系列であることに着目し LSTM を用いる方法, 2 つ目の手法としてアプリの使用履歴は class name をノードとするとグラフ構造で表現できることに着目し GCN を用いる方法を検討した。これらの特徴抽

出の有効性を評価するため, Yamamoto らの研究 [2] と同様のストレス状態の分類問題に対して分類精度を比較することで特徴抽出の性能を評価した。評価の結果, 既存手法に対して提案手法の特徴量を加えることで精度が 1.4% 向上することを確認した。

本研究では, ストレス状態推定のタスクによりアプリ使用ログの特徴量化の有効性を検証したが, 汎用性の検証には課題がある。そのため, より多くのタスクで深層学習によるアプリ使用ログの特徴量化の有効性を評価することが今後の課題として挙げられる。特に, ストレス状態推定のタスクではデータ数が限られていたため, 大規模データが利用できるタスクとの比較を行うことで, データ数に対する頑健性を評価したり, 少数データでの効率的な学習方法の検討などが課題である。

参考文献

- [1] Li, C.-T., Cao, J. and Li, T. M. H.: Eustress or Distress: An Empirical Study of Perceived Stress in Everyday College Life, *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct, UbiComp '16*, New York, NY, USA, ACM, pp. 1209–1217 (2016).
- [2] Yamamoto, N., Ochiai, K., Inagaki, A., Fukazawa, Y., Kimoto, M., Kiri, K., Kaminishi, K., Ota, J., Okimura, T., Terasawa, Y. and Maeda, T.: Physiological Stress Level Estimation Based on Smartphone Logs, *Proceedings of 2018 Eleventh International Conference on Mobile Computing and Ubiquitous Network (ICMU)*, IEEE, pp. 1–6 (2018).
- [3] Abdullah, S., Murnane, E. L., Matthews, M., Kay, M., Kientz, J. A., Gay, G. and Choudhury, T.: Cognitive Rhythms: Unobtrusive and Continuous Sensing of Alertness Using a Mobile Phone, *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '16*, New York, NY, USA, ACM, pp. 178–189 (2016).
- [4] Murnane, E. L., Abdullah, S., Matthews, M., Kay, M., Kientz, J. A., Choudhury, T., Gay, G. and Cosley, D.: Mobile Manifestations of Alertness: Connecting Biological Rhythms with Patterns of Smartphone App Use, *Proceedings of the 18th International Conference on Human-Computer Interaction with Mobile Devices and Services, MobileHCI '16*, New York, NY, USA, ACM, pp. 465–477 (2016).
- [5] Pielot, M., Dingler, T., Pedro, J. S. and Oliver, N.: When Attention is Not Scarce - Detecting Boredom from Mobile Phone Usage, *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing, UbiComp '15*, New York, NY, USA, ACM, pp. 825–836 (2015).
- [6] Phan, X.-H., Nguyen, L.-M. and Horiguchi, S.: Learning to Classify Short and Sparse Text & Web with Hidden Topics from Large-scale Data Collections, *Proceedings of the 17th International Conference on World Wide Web, WWW '08*, New York, NY, USA, ACM, pp. 91–100 (online), DOI: 10.1145/1367497.1367510 (2008).
- [7] Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural computation*, Vol. 9, No. 8, pp. 1735–1780 (1997).

- [8] Hamilton, W. L., Ying, Z. and Leskovec, J.: Representation Learning on Graphs: Methods and Applications, *IEEE Data Engineering Bulletin*, Vol. 40, pp. 52–74 (2017).
- [9] Xu, K., Hu, W., Leskovec, J. and Jegelka, S.: How Powerful are Graph Neural Networks?, *International Conference on Learning Representations*, (online), available from <https://openreview.net/forum?id=ryGs6iA5Km> (2019).
- [10] Malmi, E. and Weber, I.: You Are What Apps You Use: Demographic Prediction Based on User’s Apps., *ICWSM*, pp. 635–638 (2016).
- [11] Seneviratne, S., Seneviratne, A., Mohapatra, P. and Mahanti, A.: Predicting User Traits from a Snapshot of Apps Installed on a Smartphone, *SIGMOBILE Mob. Comput. Commun. Rev.*, Vol. 18, No. 2, pp. 1–8 (2014).
- [12] LeCun, Y., Bengio, Y. and Hinton, G.: Deep learning, *nature*, Vol. 521, No. 7553, p. 436 (2015).
- [13] Masci, J., Meier, U., Cireşan, D. and Schmidhuber, J.: Stacked convolutional auto-encoders for hierarchical feature extraction, *International Conference on Artificial Neural Networks*, Springer, pp. 52–59 (2011).
- [14] 中山英樹：深層畳み込みニューラルネットによる画像特徴抽出と転移学習，電子情報通信学会技術研究報告= IEICE technical report: 信学技報，Vol. 115, No. 146, pp. 55–59 (2015).
- [15] Kiperwasser, E. and Goldberg, Y.: Simple and accurate dependency parsing using bidirectional LSTM feature representations, *Transactions of the Association for Computational Linguistics*, Vol. 4, pp. 313–327 (2016).
- [16] Wu, C.-Y., Ahmed, A., Beutel, A., Smola, A. J. and Jing, H.: Recurrent Recommender Networks, *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining, WSDM '17*, New York, NY, USA, ACM, pp. 495–503 (online), DOI: 10.1145/3018661.3018689 (2017).
- [17] Bottou, L.: Large-scale machine learning with stochastic gradient descent, *Proceedings of COMPSTAT'2010*, Springer, pp. 177–186 (2010).