

スマートウォッチによる煩わしさを低減する 通知タイミングの推定方法に関する検討

神田琢也[†] 高橋公海[†] 宮島麻美[†]
NTT 社会情報研究所[†]

概要

健康行動を促すことを目的として、多くのヘルスケアアプリケーションで Push 通知が活用されている。Push 通知はユーザに行動を促す上で有用である一方、そのわずらわしさによってアプリの利用をやめてしまうケースも多く見られる。こういった通知の煩わしさを低減させる方法の一つとして、行動の切り替わりのタイミング (breakpoint) に通知をするという方法が挙げられる。そこで本研究では、breakpoint を検出しそのタイミングで Push 通知を送ることで煩わしさを低減させることを目指し、スマートウォッチのセンサ情報を用いて機械学習で breakpoint を推定する方法を検討し、予備実験でその有効性を検証した。

1. はじめに

高齢化社会や COVID-19 の影響により、人々のヘルスケアへの注目は近年ますます向上している。スマートホンや、スマートウォッチなどウェアラブルデバイスの普及に伴い、それらのデバイスを活用し、個人の健康管理を行う Fitbit (Fitbit Inc., San Francisco, CA, USA) ¹ や Apple Health (Apple Inc., Cupertino, CA, USA) ² , Google fit (Google Inc., Mountain View, CA, USA) ³ などに代表される健康行動を促すモバイルヘルスケアアプリケーションが多く登場している。

IQUVIA の 2021 年の調査 [1]によれば、現在 350,000 を超えるヘルスケアアプリが利用可能であり、毎日約 250 個の新しいヘルスケアアプリがリリースされている。こういったヘルスケアアプリはアプリユーザの健康管理に対する手間を削減し、健康の維持や疾病の予防に対して大きく寄与すると考えられている。

また、個人の健康管理という目的だけでなく、新しい疾病の治療手段として Digital Therapeutics (DTx) が注目されている。医薬品のように薬事承認を必要とし、医師の管理下で疾病患者自身が使用する治療目的のプログラムである。日本でもニコチン依存症改善アプリの CureApp SC (CureApp Inc., Tokyo, Japan) ⁴ や高血圧治療アプリの

CureApp HT (CureApp Inc., Tokyo, Japan) ⁵ など医療機器認定されたヘルスケアアプリが登場し注目を集めている。これらの DTx は主にスマートホンやスマートウォッチなどのデバイスを用いたアプリを活用し、疾病患者の生活習慣や行動に変化を生じさせるよう促すことにより、治療効果をもたらすことを目的としている。このように、ヘルスケアアプリで個人の生活習慣や行動に変化をもたらすことが、今後、健康状態の改善・維持においてさらに大きな役割を果たすことが予想される。

しかし、ヘルスケアアプリの課題としてユーザの利用が継続しないことやエンゲージメントの低さが指摘されている [2]。継続率を低下させる可能性のある主要な要因の一つとして、具体的にはヘルスケアアプリからの Push 通知が挙げられる。ヘルスケアアプリにおける Push 通知は、内容とタイミングが適切である場合、アプリユーザの満足度の向上に寄与するとされている [3, 4]。一方で、内容やタイミングが適切でなかった場合や、頻度が高すぎる Push 通知はユーザの満足度を低下させ、アプリを削除してしまう可能性があるとして指摘されている [4, 5]。ヘルスケアアプリでは健康に関わる情報提供や健康行動のリマインダする際に Push 通知が使われることが多く、健康行動を促す際に重要な機能の一つである。そのため、その煩わしさを低減させることは、多くのヘルスケアアプリの改善につながる事が期待できると我々は考えている。また、ヘルスケアアプリが対象としている健康行動の内容に関わらず、広い範囲のヘルスケアアプリへの応用が可能であることから通知の内容ではなく、タイミングに着目して研究を進める。

本研究では、ヘルスケアアプリの Push 通知のタイミングに着目し、スマートウォッチのセンサデータを活用して Push 通知の適切なタイミングを自動的に推定する手法を検討とする。スマートウォッチでセンシングしたデータを用いることで、従来のスマートホンなどのデバイスを活用した手法と比較して、適切なタイミングをより高頻度で検出できる可能性がある。本稿では適切な通知タイミングを推定することを目的とし、スマートウォッチのセンサ情報を用いた機械学習による推定手法の有効性の検証を行った。

Study on Estimation Method of Notification Timing by Smartwatch to Reduce Annoyance

[†]1 TAKUYA KANDA, NTT Social Informatics Laboratories.

[†]2 MASAMI TAKAHASHI, NTT Social Informatics Laboratories.

[†]3 ASAMI MIYAJIMA, NTT Social Informatics Laboratories.

1 <https://www.fitbit.com/>

2 <https://www.apple.com/jp/ios/health/>

3 <https://developers.google.com/fit/>

4 <https://sc.cureapp.com/d/>

5 <https://cureapp.co.jp/productsite/ht/>

予備実験として実験参加者 11 名の通常の業務時のスマートウォッチのセンサ情報と適切な通知タイミングと考えられる breakpoint のアノテーションデータを収集し、Long Short-Term Memory (LSTM) ネットワークを用いた機械学習モデルを構築し推定の精度の検証を行った。その結果機械学習による breakpoint の推定に対する一定の有効性を確認した。

2. 関連研究

2.1 Breakpoint を活用した通知の煩わしさの低減手法

Push 通知の煩わしさを低減させる手法の研究として、時間帯や場所、一週間のどの日か、Bluetooth で検出したユーザ間の距離、電話・SMS のログに着目した手法など多くの先行研究が存在する [6]。先行研究の中でも主要なアプローチの一つとして breakpoint に着目した手法が挙げられる。breakpoint は心理学分野における概念であり、「意味がある自然な行動が終わり、新たな行動が始まったタイミング」と定義されている [7]。この breakpoint に通知を送ることで、通知によって起きる悪影響が軽減されているという結果がいくつかの先行研究で示されている [8~15]。

Iqbal ら[8~11]はオフィス作業をさせた際にパソコン上のクリックなどのイベント情報をもとに breakpoint を推定するミドルウェアの Oasis を提案し、Oasis による breakpoint の検出とその通知の煩わしさを低減させる効果を検証した。Oasis では実験参加者の作業中の動画をもとに、その作業に関する専門家がアノテーションをすることによって breakpoint の機械学習を行っている。

Okoshi ら [12~15] はモバイル端末を対象に、breakpoint を検出する手法を提案している。これらの研究ではスマートホン操作（アプリの起動、スワイプなど）やセンサ情報をもとにした行動認識 API の認識結果をもとに breakpoint を検出する手法の提案を行っている。

Breakpoint を活用した Push 通知の煩わしさを低減させる手法は、Push 通知を breakpoint が検出されるまで遅延させることによって、Push 通知の煩わしさを低減させている。そのため、検出できる breakpoint の頻度が低いと、必要なタイミングに Push 通知を送ることができない可能性がある。

パソコンやスマートホンなど多くのデバイスで breakpoint を推定する手法が提案されている [8~15] が、これらのタイミングの推定は実験参加者がデバイスを扱っている、もしくは身に着けている状態を前提とした手法である。スマートウォッチは体に密着しているデバイスであることから、breakpoint をより高頻度で検出できる可能性がある。

スマートウォッチを活用して breakpoint の検出を行った研究としてスマートウォッチとスマートホンを併用して使

用した際の breakpoint の検出手法の検討を行った [13] が挙げられる。[13] における実験の一部としてスマートウォッチ画面のタッチなどの UI 操作と、スマートウォッチに搭載されているセンサ情報をもとにした行動認識 API を活用し「立つ」、「座る」などの行動を認識し、認識された行動の切り替わりから breakpoint を推定する手法を提案した。

2.2 先行研究と本研究の違い

本研究では breakpoint を検出しそのタイミングで Push 通知を送ることで煩わしさを低減することを目指し、スマートウォッチなどのリスト型デバイスに一般的に搭載されている加速度センサとジャイロセンサの情報をもとに、高頻度で breakpoint を検出する手法の検討を行う。

行動認識を活用した breakpoint の推定手法は「立つ」、「座る」といった行動認識 API などのシステム側が検出する対象として事前に設定された行動のカテゴリ間の切り替えタイミングで breakpoint を検出することができる。しかし、事前に設定された行動のカテゴリだけでなく、ほかの種類の行動を切り替えたタイミングでも breakpoint が発生することが想定され、これらの breakpoint の検出が可能になれば、Push 通知のタイミングを breakpoint まで遅延させる煩わしさ低減の手法が適用しやすくなると考えられる。

そこで、加速度センサやジャイロセンサから得られる情報に対して、行動認識を活用した breakpoint の検出ではなく、センサ情報に対して行動認識を活用せず、直接 breakpoint を推定する方法の検討を行う。本稿ではまず基礎検討として、機械学習による breakpoint の推定手法を構築し、予備実験でその有効性の検証を行う。

3. 実験手順

3.1 データ収集

機械学習による breakpoint の推定に用いるデータ収集のために、研究業務に従事している社員 11 名を募集し、業務中のスマートウォッチのデータと業務の様子を撮影した。実験参加者は実験の説明を受けたのち、実験参加に関する同意書にサインした。その後、利き手ではない方の手首にスマートウォッチの Fitbit Sense (Fitbit Inc., San Francisco, CA, USA) を装着し、Fitbit Sense 上に Fitbit SDK (Fitbit Inc., San Francisco, CA, USA) を用いて実装された加速度と角速度を記録するアプリを起動した。データ収集実験時の構成を図 1 に示す。

アプリの起動後参加者の作業の様子がわかる角度で動画の撮影をはじめ、通常の業務を 1 時間実施するよう依頼した。データ収集時の Fitbit に搭載された加速度と角速度のデータは Bluetooth を経由し、スマートホンの Google Pixel 5a (Google Inc., Mountain View, CA, USA) 内のサーバに送信

された。また、実験参加者が作業をしている様子の撮影には Insta 360 ONE RS (Arashi Vision Inc., Shenzhen, China) を用いた。実験中にデータ収集システムに不具合が生じた場合は、実験参加者に作業を中断してもらいデータ収集システムの再起動の実施を依頼した。

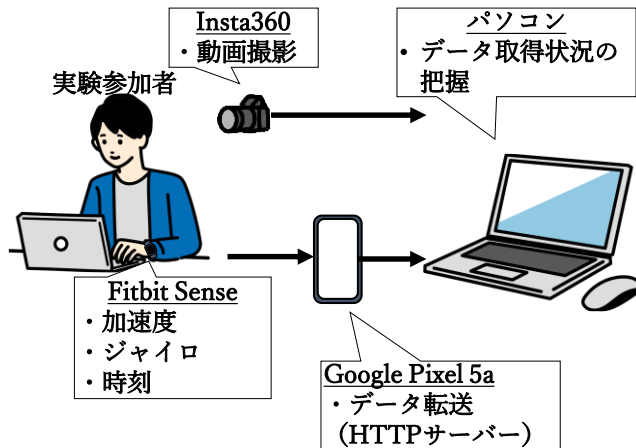


図1 データ収集実験時の構成

3.2 Breakpoint アノテーションデータの作成

各実験参加者の作業動画に対して、アノテーター2名にアノテーションデータの作成を依頼した。アノテーションデータの作成には音声・映像アノテーションツールである ELAN 5.8 [16] を用いた。アノテーターはデータ収集実験に参加しておらず、なおかつ ELAN 5.8 の操作方法に精通しているものに依頼した。アノテーターはデータ収集実験参加者の作業動画を視聴しながら、一定の間隔で分割された動画の期間 (タイムステップ) において breakpoint の発生を記録した。Iqbal ら[8~11]の実験では breakpoint の有無だけでなく、発生した breakpoint を3段階の粒度 (Course, Medium, Fine) に分類したアノテーション方法を実施したが、本稿では breakpoint の発生自体を機械学習で推定可能か検証を行うために breakpoint の有無のみを記録した。

本実験ではタイムステップの間隔を2秒と設定した。アノテーターには下記の文面でアノテーションを依頼した。「実験参加者が作業している動画を見ていただきながら Breakpoint をアノテーションして頂く作業になります。

(※Breakpoint: 意味がある自然な行動が終わり、新たな行動が始まったタイミング)。具体的には動画を見ていて実験参加者が行動を切り替えたと感じたタイミングにフラグを入れてください。どう感じたかを知りたいので、正解や間違いはありません。」

アノテーション作業完了後、アノテーターごとに判断に偏りがある可能性を考慮して、2人のアノテーターがいずれも breakpoint があると記録したタイムステップもしくは、一つずれたタイムステップに breakpoint があると記録

されたタイムステップを Breakpoint クラス、無いと記録されたタイムステップを Not Breakpoint クラスとした。



図2 ELANによるアノテーションの画面例

4. 機械学習による breakpoint 推定モデルの構築

4.1 センサデータの前処理

スマートウォッチの Fitbit Sense から得られた3軸の加速度センサ及び3軸の角速度のデータには欠損値が存在したため、前後の平均値を算出して補完した。欠損値補完処理を行った後、実験参加者の体の動きの情報をより、学習データに反映するために、加速度センサのデータに対してローパスフィルタを用いることで重力加速度の影響を除去する処理を行った。また、重力加速度の影響を除去した加速度センサから合成加速度の算出を行った。

4.2 センサデータの特徴量選択

4.1項までの前処理を施した加速度 (x, y, z), 合成加速度, 角速度 (x, y, z) のデータに対して、学習に用いる特徴量として、タイムステップごとに以下の値を算出した。

- 平均値
- 中央値
- 最大値
- 最小値
- 標準偏差

スマートウォッチ上のデータ収集アプリの不具合により、スマートウォッチのセンサ情報が記録できていないタイムステップを排除した結果、各実験参加者のデータに含まれるタイムステップ数は表1のようになった。

表 1 Not Breakpoint クラス, Breakpoint クラスの
タイムステップの数

実験参加者 ID	Not Breakpoint	Breakpoint	合計
P1	1327	16	1343
P2	1701	11	1712
P3	1256	2	1258
P4	864	2	866
P5	1504	13	1517
P6	1524	20	1544
P7	1733	5	1738
P8	1444	6	1450
P9	1491	28	1519
P10	1747	2	1749
P11	1756	14	1770
合計	16347	119	16466

4.3 機械学習モデル

スマートウォッチのセンサーデータをもとに各タイムステップの breakpoint の有無を推定するため、4.2 項までの処理を施したタイムステップごとの特徴量を入力とした、機械学習モデルを作成した。機械学習のモデルを選択する際にはロジスティック回帰や Support Vector Machine (SVM)、勾配ブースティングなど複数の手法を検討したが、最終的にテスト検証で精度の良かった Long Short-Term Memory (LSTM) ネットワークを選択した。本実験で使った LSTM の構造を表 2 に示す。作成した機械学習モデルは入力とした特徴量の次のタイムステップに breakpoint が存在する (Breakpoint クラス), 存在しない (Not Breakpoint クラス) かの 2 値分類を予測結果として出力した。入力データと予測対象のタイムステップの関係を示したものを図 3 に示す。また, Breakpoint クラスと Not Breakpoint クラスの比率に大きな偏りが生じているため, クラスごとの重みを考慮した係数をかけた学習を行った [17].

表 2 モデルの構造

入力サイズ	7×5
入力タイムステップ	30
LSTM の次元数	10
活性化関数	Sigmoid 関数
全結合層	1
最適化手法	Adam
損失関数	Binary Cross Entropy

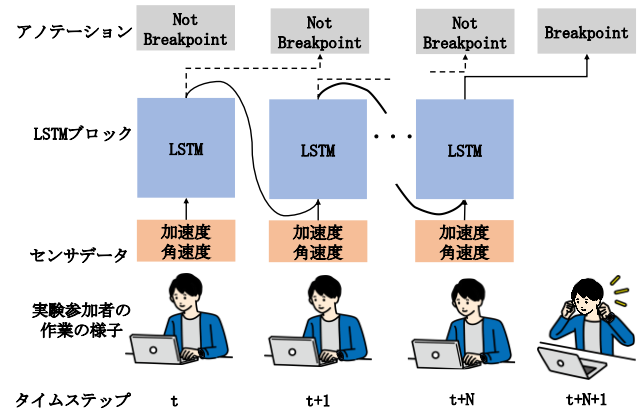


図 3 入力データと予測対象のタイムステップとの関係

4.4 評価

実験参加者 11 名のデータに対して Leave One Person Out (LOPO) 法を用いて学習を行い, モデルの推定精度を評価した。評価に用いたハイパーパラメータを表 3 に示す。

表 3 ハイパーパラメータ

バッチサイズ	32
エポック数	20
確信度の閾値	0.95

なお, 加速度, 角速度のデータが取得できていなかったタイムステップは学習時もテスト時も除外した。学習時のモデルのばらつきを考慮し, 学習モデルを 10 回作成し, それぞれの予測結果の評価指標 (Accuracy, Precision, Recall, F1 値) の平均値を算出した。表 4 に各実験参加者のデータに対する各評価指標の平均値を示す。また, 図 4 に各実験参加者のデータに対する予測結果の評価指標の箱ひげ図を示す。

表 4 評価指標の平均値

	Accuracy	Precision	Recall	F1 値
P1	1.00	0.99	0.86	0.88
P2	0.99	0.78	0.89	0.78
P3	1.00	1.00	1.00	1.00
P4	1.00	0.81	0.90	0.84
P5	0.99	0.77	0.85	0.76
P6	0.90	0.88	0.84	0.77
P7	1.00	0.64	0.78	0.65
P8	0.99	0.89	0.73	0.79
P9	0.99	0.77	0.84	0.77
P10	1.00	0.95	1.00	0.97
P11	1.00	0.90	0.86	0.87

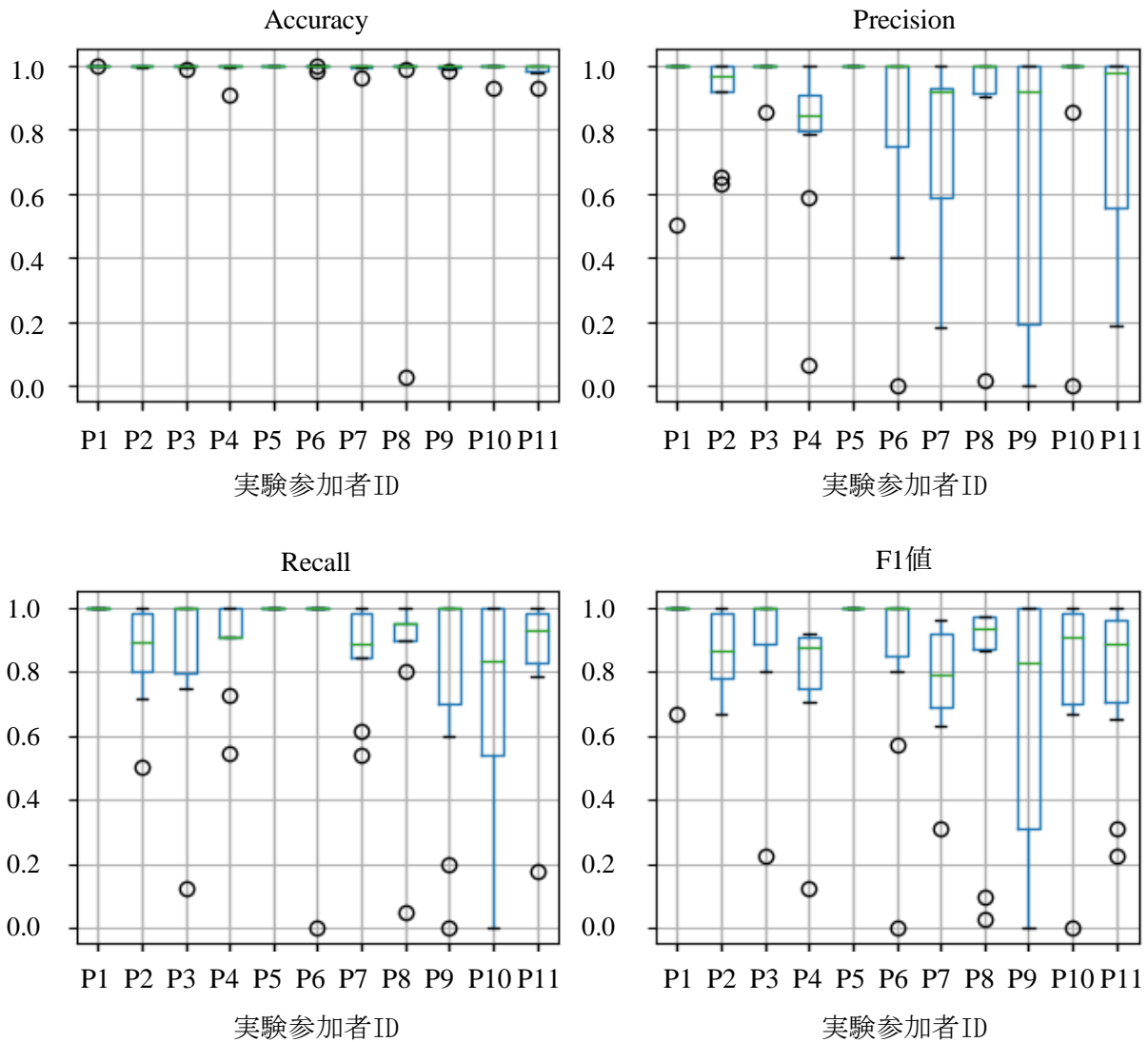


図4 各実験参加者のデータに対する予測結果 (Accuracy, Precision, Recall, F1 値)

5. 考察

5.1 Breakpoint の推定精度に関して

Accuracy は全体的に高かったが、これは Not Breakpoint クラスと Breakpoint クラスの比率に大きな偏りが生じていたためであると考えられる。Precision と Recall に関しては Accuracy ほど高い精度での予測はできなかったが、ある程度の精度で予測ができたためスマートウォッチのセンサ情報を用いた機械学習による推定手法に一定の有効性があることが示唆された。

Precision と Recall に関しては実験参加者のデータごとにも大きな差が出ている。P9 のデータに対する予測精度が Precision, Recall とともにほかの実験参加者のデータと比較して低い結果が得られた。ほかの実験参加者はデータ収集中の作業がパソコン作業であったのに対して、P9 のデータ収集実験時の作業にはパソコン作業以外の作業も多く含まれていた。今回の実験では Breakpoint クラスの学習データ

が少なかったため、LOPO 法で評価するために学習データから排除した。そのため、作業内容がほかの実験参加者のデータと大きく異なる P9 のデータに対する予測結果の精度がほかの実験参加者のデータと比較して低くなったと考えられる。

本研究は Push 通知の煩わしさを低減させることを目的としている。煩わしくないタイミングを煩わしいタイミングと誤推定しても、アプリユーザは煩わしいと感じない。一方で、煩わしいタイミングを誤って煩わしくないタイミングであると誤推定し、Push 通知をするとアプリユーザに煩わしいと感じさせてしまうことが想定される。そのため、煩わしいタイミングに通知をしないという観点から、Recall の精度の向上よりも Precision の精度の向上がより重要であると考えられる。

5.2 Breakpoint をアノテーションする作業におけるアノテーター間の個人差

本実験ではデータ収集実験で撮影した実験参加者の作業動画に対して2人のアノテーターにアノテーションを実施してもらった。各アノテーターの Breakpoint クラスと Not Breakpoint クラスの数を比較したものを表5に示す。アノテーターをそれぞれ A1, A2 と記載する。同じ実験参加者の作業動画であっても、P2, P3, P4 のアノテーションデータは各クラスのアノテーションの数が大きく異なっていた。一方で、Breakpoint クラスであるとアノテーションしているタイムステップは同一、もしくは隣り合った前後のタイムステップであった。アノテーター間で Breakpoint の数に差が生じた理由は Iqbal らの研究 [13~16] で指摘されている breakpoint の粒度による影響であると我々は考えている。アノテーションデータ作成後にアノテーションを付与した基準など感想をアノテーターに尋ねたところ、A2 のアノテーターからは「視線が大きく作業から外れた場合は breakpoint と判断した」というコメントがあり、A1 のアノテーターからは「視線が作業から外れた場合もその後作業が継続していた場合 breakpoint とは判断しなかった」とコメントがあった。これは、アノテーターごとに、breakpoint だと認識する場面に差異があるのではなく、アノテーションした際に粒度が小さいものでも Breakpoint クラスであるとアノテーションを付与したかどうかの差異であるように見受けられる。

表5 アノテーターごとの各クラスの
アノテーション数の比較

	Not Breakpoint		Breakpoint	
	A1	A2	A1	A2
P1	1318	1319	25	24
P2	1701	1667	11	45
P3	1256	1230	2	28
P4	862	829	4	37
P5	1502	1499	15	18
P6	1514	1524	30	20
P7	1733	1724	5	14
P8	1442	1431	8	19
P9	1491	1504	28	15
P10	1747	1743	2	6
P11	1756	1747	14	23
合計	16322	16217	144	249

5.3 本研究の適用範囲

本稿の実験ではオフィスのデスクワーク時の作業に関するデータを収集した検証であったため、今後はさらに生活の様々なシーンに適用範囲を広げていきたいと考えてい

る。また、本実験の参加者は普段からデスクワークを行っている研究者であり、作業の内容や実験参加者の属性として偏りが生じていたため、全てのオフィスワーカーで本手法が有効であるかどうかは実験の範囲外である点は留意する必要がある。

6. 結論・今後の展望

本稿では、ヘルスケアアプリの Push 通知の煩わしさを低減させることを目指し、適切な通知タイミングの推定に取り組んだ。Push 通知の煩わしさを低減させる手法の中でも breakpoint を活用した手法に着目し、常時アプリユーザに装着されているスマートウォッチのセンサデータを用いて LSTM ネットワークを活用した機械学習モデルで breakpoint を推定する手法を構築し、予備実験で検証を行った。

予備実験では、デスクワーク時のスマートウォッチのセンサデータの収集と breakpoint のアノテーションを行い、収集したセンサデータとアノテーションデータから LSTM ネットワークを活用したモデルで推定が可能かという検証を行った。その結果、今回の収集データでは比較的高い精度で breakpoint を推定することができたことから、スマートホンやスマートウォッチの行動認識 API などを活用せず、スマートウォッチに搭載された加速度センサやジャイロセンサの情報から直接 breakpoint を推定する手法が有効である可能性が示唆された。

今後は、さらに多く、多様な実験参加者からの作業時のデータ収集を行い検証する必要がある。また、本実験で実施した参加者は通常からデスクワークを行っている研究者であり、参加者の属性としては偏りがある。そのため、参加者の属性をより広げた実験をしたいと考えている。実験で対象とした行動に関しても、デスクワークと限られたシチュエーションであることから、さらに多様なシチュエーションでデータ収集を実施したいと考えている。また、これまでの先行研究との組み合わせを行うことで、Push 通知によってより効果的に発生する煩わしさを低減することが可能になるのではないかと考えている。

アノテーションに関しては、今回2人のアノテーターがアノテーションを行ったが、アノテーター間でアノテーションを付ける数に大きな差が発生していた。これは個人の主観に基づいてアノテーションを付けていることから、breakpoint だと判断するアノテーターの基準に差があったためであると考えられる。今後、大規模な実験や実用化を考える上では、さらに多くのアノテーションデータが必要になると考えられ、一人のアノテーターが大量の breakpoint のアノテーションデータを作成することは現実的ではないことから、複数人のアノテーターがアノテーションデータを作成することが考えられる。そのため、アノ

テーター間のアノテーションに差が生じないようにアノテーション方法を検討していきたい。

将来的には、機械学習モデルを実際にスマートウォッチ上に実装し、breakpointを検出した際にPush通知することによって、その煩わしさが低下するかを実際に検証することを検討している。また、煩わしさを低減させることによる、ヘルスケアアプリの継続利用率の変化や、健康行動を促すリマインダのPush通知をアプリ利用者が受け取った際の健康行動の実施率を調査することで、行動の変化にどの程度繋がるかを検証する必要があると考えている。

7. 参考文献

- [1] IQVIA INSTITUTE FOR HUMAN DATA SCIENCE, Digital Health Trends 2021: Innovation, Evidence, Regulation, and Adoption (online), available from <<https://www.iqvia.com/-/media/iqvia/pdfs/institute-reports/digital-health-trends-2021/iqvia-institute-digital-health-trends-2021.pdf>>, (accessed 2023-02-13).
- [2] Amagai, S., Pila, S., Kaat, A., Nowinski C. and Gershon, R.: Challenges in Participant Engagement and Retention Using Mobile Health Apps: Literature Review J Med Internet Res, doi: 10.2196/35120 (2022).
- [3] Bidargaddi, N., Almirall, D., Murphy, S., Nahum-Shani I, Kovalcik, M., Pituch, T., Maaich, H. and Strecher, V.: To Prompt or Not to Prompt? A Microrandomized Trial of Time-Varying Push Notifications to Increase Proximal Engagement With a Mobile Health App, JMIR Mhealth Uhealth, doi: 10.2196/10123 (2018).
- [4] Haggag, O., Grundy, J., Abdelrazek, M. and Haggag, S.: A large scale analysis of mHealth app user reviews. Empir Softw Eng, doi: 10.1007/s10664-022-10222-6 (2022).
- [5] Morrison, L.G., Hargood, C., Pejovic, V.: Geraghty AWA, Lloyd S, et al., The Effect of Timing and Frequency of Push Notifications on Usage of a Smartphone-Based Stress Management Intervention: An Exploratory Trial., PLOS ONE, doi: <https://doi.org/10.1371/journal.pone.0169162>, (2017).
- [6] Abhinav, M., Musolesi, M.: Intelligent notification systems: A survey of the state of the art and research challenges, arXiv preprint arXiv:1711.10171, (2017).
- [7] Newton, D. and Engquist, G.: The perceptual organization of ongoing behavior, Journal of Experimental Social Psychology, vol. 12, no. 5, pp. 436–450 (1976).
- [8] Iqbal, S. T. and Horvitz, E.: Notifications and awareness: A field study of alert usage and preferences, Proc. 2010 ACM Conference on Computer Supported Cooperative Work (CSCW '10), pp. 27–30, (2010).
- [9] Iqbal, S. T. and Bailey, B. P.: Understanding and developing models for detecting and differentiating breakpoints during interactive tasks, Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '07), pp. 697–706 (2007).
- [10] Iqbal, S. T. and Bailey, B. P.: Investigating the effectiveness of mental workload as a predictor of opportune moments for interruption, Proc. 05 Extended Abstracts on Human Factors in Computing Systems (CHI EA '05), pp. 1489–1492 (2005).
- [11] Iqbal, S. T. and Bailey, B. P.: Leveraging characteristics of task structure to predict the cost of interruption, Proc. SIGCHI Conference on Human Factors in Computing Systems (CHI '06), pp. 741–750 (2006).
- [12] Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A. K. and Tokuda, H.: Attelia: Reducing user's cognitive load due to interruptive notifications on smart phones, Proc. 2015 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, pp. 96–104 (2015).
- [13] Okoshi, T., Ramos, J., Nozaki, H., Nakazawa, J., Dey, A. K. and Tokuda, H.: Reducing users' perceived mental effort due to interruptive notifications in multi-device mobile environments, Proc. 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp), pp. 475–486 (2015).
- [14] Okoshi, T., Tsubouchi, K., Taji, M., Ichikawa, T. and Tokuda, H.: Attention and engagement-awareness in the wild: A large-scale study with adaptive notifications, Proc. 2017 IEEE International Conference on Pervasive Computing and Communications (PerCom), IEEE, pp. 100–110 (2017).
- [15] Okoshi, T., Tsubouchi, K. and Tokuda, H.: Realworld product deployment of adaptive push notification scheduling on smartphones, Proc. the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data mining (KDD), pp. 2792–2800 (2019).
- [16] Brugman, H., Russel, A.: Annotating multi-media/multimodal resources with ELAN, Proc. the Fourth International Conference on Language Resources and Evaluation (LREC'04) (2004).
- [17] King, G., Zeng, L.: Logistic regression in rare events data. Political analysis, 9(2), pp.137-163 (2001).