

ウェブサイトからマウス履歴を取得するセッションリプレイサービスの検出ツールの開発

田口 凱之† 菊池 浩明†

明治大学総合数理学部†

1 はじめに

セッションリプレイサービスは、利用者がウェブページ上で行うマウス操作やクリック、スクロールといった行動を記録・解析し、サイト改善やマーケティングに役立てる技術である。しかし、その一方で、ユーザ側からはこうしたサービスの存在を直接確認することは難しく、プライバシー面での不安が懸念されている。梶間らの研究 [1] によると、多くのユーザは自分のウェブ上で行動が外部の追跡サービスによって記録・利用されていることを十分に認識していないと報告されている。

そこで、本研究では、ウェブサイト導入されているセッションリプレイサービスを自動的に検出するツール（以後検出ツールと呼ぶ）を開発し、その精度を評価した。まず自分で用意したウェブサイトにてセッションリプレイサービスを仕込み、その挙動を確認した上で、検出手法を提案・実装した。さらに、実際の主要なウェブサービスを対象に検出精度を評価し、提案手法の有効性を検証した。

2 セッションリプレイサービスの導入状況調査

2.1 調査方法

ウェブサイト上のトラッキングスクリプトを検出し、サイトが使用しているトラッキング技術やツールを排除するブラウザ拡張である、Ghostery[3]を用いて、ウェブサイト導入されているセッションリプレイサービスを手動で調査した。

2.2 調査対象

トップウェブサイトランキングを提供しているTranco[4]の2024年のデータセットの内、事前にページが閲覧可能な世界の上位70サイト、および、日本の上位70サイトのリストを調査対象とした。

検出対象のセッションリプレイサービスは、Microsoft Clarity^{*}、Hotjar[†]、Mouseflow[‡]、Yandex Metrika[§]、

表1 トップウェブサイトにおけるセッションリプレイサービスの導入状況（全サイト数を母数）

サービス名	世界 (70 サイト)		日本 (70 サイト)	
	数	%	数	%
Microsoft Clarity	12	17.1	12	17.1
Hotjar	2	2.9	6	8.6
Mouseflow	0	0.0	0	0.0
Yandex Metrika	2	2.9	0	0.0
Contentsquare	0	0.0	0	0.0
Crazyegg	1	1.4	1	1.4
Dynatrace	0	0.0	1	1.4
FullStory	1	1.4	3	4.3
LogRocket	0	0.0	0	0.0
Lucky Orange	0	0.0	0	0.0
合計	18	25.7	23	32.9

contentsquare[¶]、Crazyegg^{||}、Dytatrace^{**}、fullstory^{††}、luckyorange^{‡‡}、logrocket^{§§}の10である。

2.3 調査結果

表1にセッションリプレイサービスの導入率を示す。日本のサイトの方が導入が進んでいる。世界の上位70サイトに載っているサイトの多くが日本の上位70サイトにも載っていることを考慮すると、世界の上位を占めるトップウェブサイトにおいて、それほどセッションリプレイサービスの導入が進んでいないことが推察される。

3 セッションリプレイサービス検出ツールの開発

3.1 概要

本研究では、Web ページ上に埋め込まれているセッションリプレイサービス特有のコードパターンを自動的に抽出し、当該サイトがいずれかのセッションリプレイツールを導入しているかを判別するシステムを開発した。本システムでは、以下の原理と手順に基づいて検出する。

まず、ウェブサイトにてセッションリプレイサービスを仕込み、その挙動を確認した。図1に、取得したマウス

Development of Web Session Replay Service Detection Tool

†Kaiji Taguchi, Hiroaki Kikuchi, School of Interdisciplinary Mathematical Science, Meiji University.

*<https://clarity.microsoft.com/>

†<https://www.hotjar.com/>

‡<https://mouseflow-jp.com/>

§<https://www.crazyegg.com/>

¶<https://contentsquare.com/>

||<https://www.luckyorange.com/>

**<https://www.dynatrace.com/ja/>

††<https://www.fullstory.com/>

‡‡<https://www.luckyorange.com/>

§§<https://logrocket.com/>

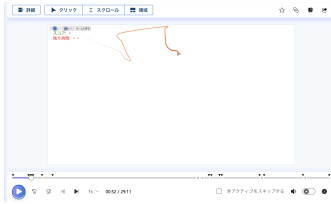


図1 デモ画面例

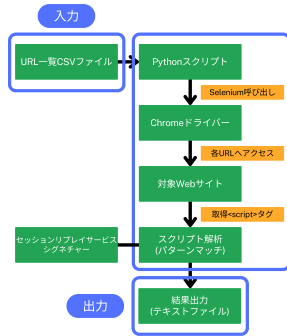


図2 システム構成図

履歴のデモ画面を示す。

セッションリプレイサーはサービス提供者ごとに異なる JavaScript コードや特定のスクリプト URL を用いている。本システムはこれらサービス固有のコードパターン（例えば、`clarity.ms/tag/` など）を定義しておく。対象サイトのソースコード中にこれらのパターンが出現すれば、対応するセッションリプレイサーが導入されていると判定可能する。

処理の流れは以下の通りである。

- (1) 事前に用意した CSV ファイルから解析対象サイトの URL リストを読み込む。
- (2) Selenium WebDriver と ChromeDriver を用いて自動的に各サイトへアクセスし、取得した HTML から `<script>` タグを抽出する。
- (3) 本システムは、各 `<script>` 要素の `src` 属性や内部コードを正規表現マッチングにより解析し、定義済みのパターンが検出された場合は対応するサービス名を出力する。

本システムは Python を用いて実装している。システム構成図を図2に示す。また、本研究では Ghostery による検出結果を参考に正解データを定め、本システムの同一サイト群に対する検出結果の精度を評価した。指標は Precision（適合率）、Recall（再現率）、F1 値を評価する。

3.2 結果

検出ツールの精度を表2に示す。実験の結果、同じサイトに複数のセッションリプレイサーが導入されているケースにおいて、サービスごとに区別すると FN については世界で8件、日本で16件であった。また、FPについては世界で3件、日本で2件であった。

3.3 誤検出、未検出の例と原因の考察

FN となったサイトの例としては、<https://www.bing.com/> や <https://unity.com/ja> などが挙げられる。このようなサ

表2 検出ツールの精度指標

	Precision	Recall	F1
世界	0.769	0.556	0.645
日本	0.778	0.304	0.438
平均	0.774	0.43	0.542

```

    view-source:https://wordpress.com/ja/
    window.clarity =
    window.clarity ||
    function () {
    ( window.clarity.q = window.clarity.q || [] ).push( arguments );
    };
    const clarityScript = kit.attachScriptElement( 'https://www.clarity.ms/tag/j8c1i1d8a' );
    document.body.appendChild( clarityScript );
    
```

図3 wordpress.com のソースコードの一部

イトに対して手動でトップページのソースコードを調べたところ、定義済みのパターンと一致する文字列は見受けられなかった。原因としては、トップページ以外のページにセッションリプレイサーが導入されている場合や、本実験で使用したパターンと異なる JavaScript コードや特定のスクリプト URL を用いている場合などが考えられる。

FP となったサイトの例として、<https://wordpress.com/ja/> を挙げる。このサイトに対して手動でトップページのソースコードを調べたところ、図3のように定義済みのパターンと一致する文字列 (`clarity.ms/tag/`) が見つかった。よって検出ツールでは Microsoft Clarity が導入されていると判定したが、Ghostery ではいずれのセッションリプレイサーも検出されなかった。原因として、この Clarity は昔は使われていたが、現在は動作していない可能性が考えられる。

4 おわりに

本研究では、国内外の主要なウェブサイトにおけるセッションリプレイサーの導入状況を調査し、セッションリプレイサー検出ツールを開発した。平均 F1 で 0.542 の精度を評価した。今後の課題として、まず、実験で使用した URL リストを対象に、各ウェブサイトで実際にセッションリプレイサーが利用されているかを手動で確認する作業を挙げる。そして得られた結果を基準データとし、これをもとに、Ghostery と本研究で開発した検出ツールの検出結果を比較し、それぞれの精度を評価する。さらに、本ツールの精度が低い原因を詳しく調査し、改善点を特定する。

参考文献

- [1] 梶間大地, 菊池浩明, “セッションリプレイサーからの個人識別性と国内外サイトにおけるプライバシーポリシーでの公表状況”, 第104回 CSEC 研究発表会, 2024-CSEC-104, No.11, pp1-8, IPSJ, 2024.
- [2] Selenium, (<https://www.selenium.dev/ja/documentation/>, 2024年8月参照).
- [3] Ghostery, (<https://www.ghostery.com/>, 2024年8月参照).
- [4] Tranco, (<https://tranco-list.eu/>, 2024年8月参照).