ECサイトに潜むソーシャルエンジニアリング：ローカルSLMを用いたダークパターン検知

橋本　拓也†　　服部　　峻††　　宮城　茂幸††

† 滋賀県立大学大学院　〒 522–8533 滋賀県彦根市八坂町 2500
†† 滋賀県立大学先端工学研究院　〒 522–8533 滋賀県彦根市八坂町 2500
E-mail: †to23thashimoto@ec.usp.ac.jp, ††{hattori.s,miyagi.s}@e.usp.ac.jp

あらまし　近年，デジタルサービスにおいてユーザの意思決定を不当に操作し不利益をもたらす UI デザインである「ダークパターン」が社会問題化している．これに対し，大規模言語モデル (LLM) を用いた自動検知手法の研究が進められているが，既存手法の多くは API 経由で利用する SaaS 型 LLM に依存しており，コストや再現性の観点から大規模な実態調査への適用が困難である．そこで本研究では，ローカル環境で動作する小規模言語モデル (SLM) を活用した，低コストかつ自律的なダークパターン検知手法を提案する．提案手法では，レンダリング結果から視覚情報を抽出し HTML 属性として埋め込む「視覚情報埋め込みを伴う構造保持型 DOM 分割」アルゴリズムを導入することで，軽量なモデルでも文脈を考慮した解析を目指す．本稿では，日本の EC サイトを対象に収集したデータセットに対し，4bit 量子化された Qwen2.5-Coder-7B を用いた評価実験を行い，VRAM が 16GB 程度の環境における提案手法の有効性と実用性を検証する．

キーワード　ダークパターン，EC サイト，UI，SLM

1　はじめに

現代のデジタル社会において，Web サービスやアプリケーションにおける UI (User Interface) および UX (User Experience) 設計の重要性が高まっている．洗練された UI/UX はユーザビリティを向上させ，ユーザに利益をもたらすが，その一方で事業者の利益を優先し，ユーザに不利益な意思決定を意図的に誘導する設計手法への懸念も広がっている．このような設計は「ダークパターン (Dark Patterns)」と呼ばれ，2010 年に Harry Brignull によって提唱された概念である．Brignull はこれを「ユーザーを騙して，たとえば，品物の購入時に保険に入らせたり，定期購入を契約させたりなど，特定の行動に誘導するため慎重に設計されたユーザーインターフェース [1]」と定義している．なお，近年では「ディセプティブパターン (Deceptive Patterns)」という呼称も提唱されているが，本稿では広く定着している「ダークパターン」という用語を用いて議論を進める．

ダークパターンの具体例としては，視覚的階層操作によって特定の選択肢を誤認させる「Interface Interference」，虚偽のカウントダウンタイマー等を用いてユーザの焦燥感を煽る「Social Engineering」，サービスの解約プロセスを意図的に複雑化する「Obstruction」などが挙げられる．これらの設計は，金銭的な損失やプライバシーの侵害といった消費者トラブルに直結することから，重大な社会問題として認識されつつある．

現在，欧米諸国を中心にダークパターンに対する法規制の議論が進展しており [2]，日本国内においても対策の必要性が叫ばれている．実効性のある規制や対策を講じるためには，Web 上におけるダークパターンの蔓延状況を把握する大規模な実態調査が不可欠である．しかし，人手による調査には限界がある上，現時点で実用的なダークパターン自動検知手法は確立しておらず，大規模調査を実施した事例は限られている．この問題を解消するためには，高精度かつ低コストな自動検知技術の確立が急務である．

近年の LLM (Large Language Model) 技術の飛躍的な進歩に伴い，LLM を用いたダークパターン検知の研究が注目を集めている．これらの研究では，従来の古典的な手法と比較して高い検出精度が報告されている．しかし，先行研究の多くは，API 経由で利用する SaaS (Software as a Service) 型の LLM に依存しており，実運用におけるいくつかの課題が浮き彫りになっている．第一にコストの問題である．SaaS 型 LLM の多くは従量課金制を採用しており，大規模なクローリングデータに対して全件検査を行う場合，API 利用料が莫大となり予算的な障壁となる．第二に再現性と持続可能性の問題である．SaaS 型モデルは頻繁にアップデートや仕様変更が行われるため，同一の入力に対しても時期によって出力が異なる可能性があり，学術的な再現性の担保が困難である．また，サービス終了やポリシー変更による利用停止のリスクも無視できない．

そこで本研究では，SaaS 型 LLM に依存しない，ローカル環境で動作する SLM (Small Language Model) に着目する．本研究の目的は，ローカル SLM を用いた低コストかつ自律的なダークパターン検知手法を確立し，大規模な実態調査を技術的に支援することである．具体的には，収集した Web ページのソースコードに対し，パラメータ数を抑えた軽量なモデルを用いて解析を行う．ローカル SLM を採用することによる主な利点は以下の通りである．

- **コスト効率**: API 利用料が発生しないため，トークン数を気にすることなく大規模なデータを解析可能である．
- **再現性と透明性**: モデルをローカル環境で管理するため，

バージョンの固定が可能であり，外部要因による実験結果の変動を防ぐことができる．

本稿では，実際に日本語 EC サイトから収集したデータセットを用いて，提案手法であるローカル SLM によるソースコード解析の有効性を検証する．

本稿の構成は以下の通りである．第 2 章では，関連研究について述べ，第 3 章では，視覚情報の埋め込みを伴う前処理とローカル SLM による推論を組み合わせた提案手法について詳述する．第 4 章では，構築したシステムを用いた評価実験の結果と考察を述べ，第 5 章で本稿の結論と今後の展望をまとめる．

2 関 連 研 究

ダークパターンに関する研究は，主に分類と検知の 2 つの領域に焦点が当てられている．本章では，これら 2 つの領域の関連研究について述べる．

2.1 ダークパターン分類法

2024 年，Gray ら [3] は，既存の 10 種類のダークパターンに関する規制上および学術上の分類を調和させ，高レベル，中レベル，低レベルの 3 階層に分類される 64 種類の統合ダークパターンタイプについて，標準化された定義を備えたオントロジーを提案した．

10 種類のダークパターンの分類の詳細は以下の通りである．

- 2010 年から Harry Brignull 自身の WEB サイト [1,2]で共有されている分類
- 学術分野におけるダークパターン分類 4 件
- EU，英国，米国の利害関係者や規制当局のダークパターン分類を含む公開文書 5 件

これらの分類法に含まれるダークパターンタイプを分析し，最終的に高レベルの分類 5 種類，中レベルの分類 25 種類，低レベルの分類 35 種類，計 64 種類のオントロジーとされている．

Gray らによるオントロジーのうち，高レベルのダークパターンタイプとその定義を以下に示す．なお，日本語訳は著者による．

- **Sneaking** is a strategy which hides, disguises, or delays the disclosure of important information that, if made available to users, would cause a user to unintentionally take an action they would likely object to.
  （**スニーキング**とは，重要な情報を隠す，偽装する，開示を遅らせる戦略で，もしその情報がユーザに提示されていれば拒否したであろう行動を，ユーザに意図せずとらせるもの.）
- **Obstruction** is a strategy which impedes a user's task flow, making an interaction more difficult than it inherently needs to be, dissuading a user from taking an action.
  （**妨害**とは，ユーザのタスク進行を阻害し，本来必要とされる以上にインタラクションを困難にすることで，ユーザーが特定のアクションをとることを思いとどまらせる戦略.）
- **Interface Interference** is a strategy which privileges specific actions over others through manipulation of the user interface, thereby confusing the user or limiting discoverability of relevant action possibilities.
  （**インターフェース干渉**とは，UI の操作を通じて特定のアクションを他のアクションよりも優遇し，それによってユーザを混乱させたり，関連するアクションの選択肢の発見可能性を制限したりする戦略.）
- **Forced Action** is a strategy which requires users to perform an additional and/or tangential action or information to access (or continue to access) specific functionality, preventing them from continuing their interaction with a system without performing that action.
  （**強制されたアクション**とは，特定の機能にアクセス（またはアクセスを継続）するために，追加的または付随的アクションの実行や情報の提供をユーザに要求し，そのアクションを実行しない限り，システムとの対話を継続できないようにする戦略.）
- **Social Engineering** is a strategy which presents options or information that causes a user to be more likely to perform a specific action based on their individual and/or social cognitive biases, thereby leveraging a user's desire to follow expected or imposed social norms.
  （**ソーシャルエンジニアリング**とは，個人的または社会的な認知バイアスに基づき，ユーザが特定のアクションを実行する可能性を高めるような選択肢や情報を提示する戦略. これは，期待される，あるいは課された社会規範に従いたいというユーザの欲求を利用するものである.）

2.2 ダークパターン検知

ダークパターン検知の研究は，近年の情報処理技術の急速な発展により，その検知手法が大きく変化してきている．

2022 年，矢田ら [4] は EC サイトを対象に，正解データが付与された自然言語テキストのデータセットを用いて，機械学習によるダークパターン自動検出のベースライン評価を行った．このベースライン評価では，ダークパターンの有無を 2 値分類で検出を行い，評価が行われた．

2023 年，Mansur ら [5] は，スクリーンショットを入力とし，画像処理技術および自然言語処理技術を用いてダークパターンを検知するシステムである AidUI を提案した．AidUI は，パターンマッチングを用いたテキスト分析，カラーヒストグラム分析技術を用いた色分析，隣接するセグメントのサイズを計算することによる空間分析によって，ダークパターンの検出を行っている．

2025 年，Zewei Shi ら [6] はマルチモーダル LLM を用いてダークパターンを自動検知する DPGuard を提案した．DPGuard は，ダークパターンの有無を判別するバイナリ分類器と，マルチモーダル LLM を用いたダークパターンのカテゴリを出

[1] 旧：https://darkpatterns.org
[2] 現：https://www.deceptive.design

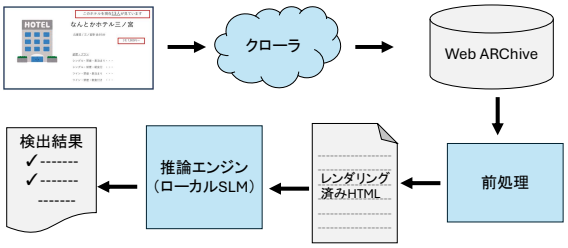

図 1　提案手法 構成図

力するダークパターン検出器からなる．これは，バイナリ分類器でダークパターンを含むと判断されたもののみマルチモーダル LLM を用いた検出器で検知を行うことにより計算コストの削減を図っている．

3　提 案 手 法

本稿では，日本語 EC サイトを対象にスクレイピングを行い，収集した WEB サイトデータを対象にローカル SLM を用いてソースコード解析を行うことにより，ダークパターンの検知を試みる．本提案手法は，スクレイピング，前処理，検知の 3 フェーズに分けられる．

本稿では，2024 年に Gray ら [3] が提案したダークパターンの分類法を採用する．しかし，すべてのダークパターンを対象に実験を行うことは難しい．例えば，カートの画面や購入手続きを進めた際に出現するダークパターンは，そのページの取得が困難である．そこで，高レベルの 5 分類のうち，Social Engineering を対象に実験を行う．これに含まれるダークパターンタイプを表 1 に示す．本稿では，Social Engineering に含まれる 8 タイプについて，検知を試みる．

また，本稿では VRAM が 16GB 程度の環境を想定しており，システムの検証環境として Google Colab(GPU:T4, VRAM:15GB)，本番環境としてデスクトップ PC(GPU:RTX 5060 ti, VRAM:16GB) を使用する．

3.1　スクレイピング

近年の WEB サイトは，JavaScript などを用いて動的に構築されるケースが多い．このため，動的なスクレイピングが可能な BrowsertrixCrawler[3]を採用する．Browsertrix Crawler とは，ブラウザベースのクロールシステムであり，Docker コンテナで実行できるよう設計されている．これにより取得された WARC ファイルを入力として用いる．

実際の EC サイトに対してスクレイピングを行うにあたり，EC サイトの URL のリストが必要である．そこで，日本語 EC サイト 200 件の URL を手動で記録した．なお，これらの EC サイトは，

- 2025 ネット通販売上高ランキング TOP100（日本ネット経済新聞）
- Shopify 導入事例集
- BASE 導入事例

より取得した．この 200 件の EC サイトに対し，Browsertrix Crawler を用いて 1 サイトあたり 15 ページのスクレイピングを行った．保存されたページは，200 サイト計 2666 ページである．

3.2　前　処　理

スクレイピングにより取得された WARC (Web ARChive) ファイルは，その状態では SLM の入力に適さない．よって，SLM への入力に適した形式に変換する必要があり，このための前処理機構を作成する．

Browsertrix Crawler を用いて収集した WARC ファイルには，HTML のほかに CSS や JavaScript，画像などが含まれる．本稿では，取得した WARC ファイルをヘッドレスブラウザでレンダリングし，レンダリング済み HTML として保存する．これにより，スクリーンショットを用いずに視覚的情報の取得を試みる．

具体的には，ヘッドレスブラウザ上で JavaScript を実行し，動的なコンテンツが展開された状態の DOM (Document Object Model) ツリーを構築する．この際，各 DOM ノードに対して window.getComputedStyle を適用し，CSS 適用後の計算済みスタイルを HTML 属性として明示的に埋め込む処理を行う．例えば，display: none や opacity: 0 といった非表示スタイルを持つ要素には不可視属性を，ボタンやリンク等の重要要素には背景色やフォントサイズを属性値として付与する．これにより，テキストベースのモデルであっても，「視覚的な隠蔽」や「配色の強調による誘導」といった視覚的特徴を DOM 構造の一部として認識可能にする．

続いて，SLM の限られたコンテキスト長を有効活用するため，意味論的浄化を行う．具体的には，ダークパターンの判定に寄与しない <script>, <style>, <svg> タグや HTML コメント等を削除し，DOM 構造を軽量化する

最後に，軽量化された DOM ツリーを SLM の入力制限（本研究では 2048 トークンと設定）に合わせて分割する．単純な文字列分割ではタグの整合性が崩れ，階層構造の欠落によりモデルが文脈を見失うリスクがある．そのため，本手法では分割された各チャンクの先頭に，ルート要素から当該ノードに至るまでの祖先タグ列（パンくずリスト）を自動的に挿入する．これにより，局所的なチャンクであってもページの全体構造における位置と文脈を保持したまま，推論を行うことが可能となる．

3.3　検　　知

3.4　検知モデルの選定と最適化

本稿では，検知に用いるローカル SLM として，Alibaba Cloud によって開発されたコーディング特化型モデルである

[3] https://crawler.docs.browsertrix.com/

表 1　対象とするダークパターンタイプ

Type	Definition
High Demand	Indicates that a product is in high-demand or likely to sell out soon, even though that claim is misleading or false.
Low Stock	Indicates that a product is limited in quantity, even though that claim is misleading or false.
Endorsements and Testimonials	Indicates that a product or service has been endorsed by another consumer, even though the source of that endorsement or testimonial is biased, misleading, incomplete, or false.
Parasocial Pressure	Indicates that a product or service has been endorsed by a celebrity, influencer, or other entity that the user trusts, even though the source of that endorsement is biased, misleading, incomplete, or false.
Activity Messages	Describes other user activity on the site or service, even though the data presented about other users' purchases, views, visits, or contributions are misleading or false.
Countdown Timer	Indicates that a deal or discount will expire by displaying a countdown clock or timer, even though the clock or timer is completely fake, disappears, or resets automatically.
Limited Time Message	Indicates that a deal or discount will expire soon or be available only for a limited time, but without specifying a specific deadline.
Confirmshaming	Frames a choice to opt-in or opt-out of a decision through emotional language or imagery that relies upon shame or guilt.

Algorithm 1 視覚情報埋め込みを伴う構造保持型 DOM 分割

Require: WARC records $\mathcal{W}$, Target URLs $\mathcal{U}$, Max tokens L_{max}
Ensure: Set of structured chunks $\mathcal{D}$

$\mathcal{D} \leftarrow \emptyset$
for each record $r \in \mathcal{W}$ **do**
  if $r.url \in \mathcal{U}$ **and** $r.status = 200$ **then**
    $h \leftarrow \text{HeadlessRender}(r.content)$
    $h' \leftarrow \text{InjectVisualFeatures}(h)$ ▷ Inject computed styles as attributes
    $t \leftarrow \text{CleanDOM}(h')$ ▷ Remove scripts, styles, comments
    $C \leftarrow \text{RecursiveChunking}(t.body, \emptyset, L_{max})$
    $\mathcal{D} \leftarrow \mathcal{D} \cup C$
  end if
end for
return $\mathcal{D}$
procedure RecursiveChunking($node, ancestors, L_{max}$)
  $S_{start} \leftarrow \text{StartTag}(node)$
  if $\text{Length}(buffer) + \text{Length}(S_{start}) > L_{max}$ **then**
    $\text{Flush}(buffer)$
    $buffer \leftarrow \text{Breadcrumbs}(ancestors)$ ▷ Initialize with ancestor tags
  end if
  Append S_{start} to $buffer$
  for each $child \in node.children$ **do**
    $\text{RecursiveChunking}(child, ancestors \cup \{node\}, L_{max})$
  end for
  Append $\text{EndTag}(node)$ to $buffer$
end procedure

Qwen2.5-Coder [7] を採用する．本モデルは，自然言語に加え HTML や CSS などのマークアップ言語の理解に優れており，DOM 構造の解析に適している．モデルのバリエーションには，パラメータ数が 0.5B, 1.5B, 3B, 7B, 14B, 32B のものが存在する．

モデルの選定にあたり，本研究の検証環境である Google Colab（NVIDIA T4 GPU, VRAM 15GB）において，動作確認と VRAM 使用量の検証を行った．検証の結果，14B モデルでは各パラメータを 4bit に丸める量子化（4-bit Quantization [8]）を適用しても，長い HTML コンテキストを入力した際にメモリ不足（OOM）が発生するリスクが高いことが確認された．一方，7B モデルに対し 4bit 量子化（NF4 形式）を適用した場合，モデル重みのメモリ消費を約 5.5GB に抑制できることが判明した．これにより，残りの VRAM 領域を KV キャッシュやアクティベーション計算に割り当てることが可能となり，最大シーケンス長 4096 トークンかつバッチサイズ 2 での並列推論が安定して動作することを確認した．

以上の検証より，本研究では計算資源の制約下で処理速度とコンテキスト容量のバランスが最適である “Qwen2.5-Coder-7B-Instruct” の 4bit 量子化モデルを採用することとした．また，推論エンジンの実装には ‘Unsloth’ ライブラリを用い，メモリ効率化と推論速度の向上を図っている．

4 評 価 実 験

提案手法の有効性を検証するために実施する評価実験について述べる．本実験の目的は，計算資源に制約のある環境において，4bit 量子化されたファインチューニング無しの SLM を用いたソースコード解析が，Web 上のダークパターン検知において実用的な精度と網羅性を持つかを定量的に明らかにすることである．

実験は，データ収集，前処理，モデルによる推論，および人間の評価者による正解の付与というプロセスを経て行われる．

4.1 実験用データセットの構築と前処理

評価実験の基礎となるデータセットには，独自にクローリングを行い収集した EC サイトの WARC ファイルのうち，2025 ネット通販売上高ランキング TOP100 の 100 サイトより収集

Algorithm 2 SLM を用いたダークパターン検知

Require: Structured chunks $\mathcal{D}$, Taxonomy definitions $\mathcal{T}$, Target category C_{tgt}, Batch size B, Model $\mathcal{M}$ (4-bit quantized)
Ensure: Set of detected patterns $\mathcal{P}$

$\mathcal{P} \leftarrow \emptyset$
$\delta \leftarrow \mathcal{T}[C_{tgt}]$ ▷ Retrieve specific definition to minimize context
Filter $\mathcal{D} \leftarrow \{d \in \mathcal{D} \mid \text{Length}(d.html) \geqq L_{min}\}$
Partition $\mathcal{D}$ into batches $\mathcal{B} = \{b_1, b_2, \ldots, b_m\}$ where $|b_i| \leqq B$
for each batch $b \in \mathcal{B}$ **do**
  $inputs \leftarrow \emptyset$
  for each chunk $d \in b$ **do**
    $p \leftarrow \text{ConstructPrompt}(\delta, d.context, d.html)$
    Append p to $inputs$
  end for
  $X \leftarrow \text{Tokenize}(inputs, \text{padding} = \text{True})$
  $Y \leftarrow \mathcal{M}.\text{Generate}(X)$ ▷ Batch inference on GPU
  for each output $y \in Y$ **do**
    $J \leftarrow \text{ExtractJSON}(y)$
    if $J \neq \text{null}$ **and** $J.\text{patterns} \neq \emptyset$ **then**
      $\mathcal{P} \leftarrow \mathcal{P} \cup J.\text{patterns}$
    end if
  end for
  ClearGPUMemory() ▷ Free VRAM to prevent fragmentation
end for
return $\mathcal{P}$
procedure ConstructPrompt($\delta, context, html$)
  $S_{sys} \leftarrow \text{FormatSystemPrompt}(\delta)$
  $S_{user} \leftarrow$ "Context: " + $context$ + \n + "HTML: " + $html$
  return $\text{ApplyChatTemplate}(S_{sys}, S_{user})$
end procedure

したデータからランダム抽出した 200 ページを実験用データセットとし，これを対象とした．収集したデータを SLM で解析を行うため，前章で提案した「視覚情報埋め込みを伴う構造保持型 DOM 分割」アルゴリズムを適用する．

具体的には，まずヘッドレスブラウザを用いて各ページをレンダリングし，ユーザーが視認するスタイル情報を HTML 属性として明示的に埋め込む．次に，スクリプトや広告などのノイズを除去した後，トークン制限（L_{max}）に基づき，DOM の階層構造とパンくずリスト（Context Path）を保持したまま，HTML を意味のある単位（チャンク）へと分割する．これらの処理済みデータは，メタデータと共に JSON 形式で保存し，推論用データセットとした．

4.2 推論設定とプロンプト設計

推論モデルには Qwen2.5-Coder-7B-Instruct を採用した．コンシューマー向けの限られた GPU メモリ（VRAM 16GB）環境下での推論を可能にするため，Unsloth ライブラリを用いてモデルの重みに 4bit 量子化（NF4 形式）を適用した．推論パイプラインは，バッチサイズを 2，チャンクあたりの入力最大トークン長を 4096 に設定し，並列処理によるスループットの最大化を図った．

表 2 推論パフォーマンスの比較

バッチサイズ	総処理時間	処理速度	VRAM 使用量 (GB)	
	(s)	(sec/file)	平均	最大
2	15166	75.83	6.9	7.1
4	11837	59.18	8.3	9.1
8	10003	50.01	11.3	12.7

プロンプトを記述する言語は英語とし，システムプロンプトではモデルの役割およびタスクを指示した．また，出力フォーマットは抽出要素・パターン名・推論理由を含む JSON 形式で出力するように指示した．なお，ダークパターンが存在しない場合はからのリストを出力するよう指示した．

ダークパターンの定義プロンプトは，パターンの定義文のみのプロンプト A，定義文に加え例及および除外条件を加筆したプロンプト B を用意し，それぞれで実験を行った．定義文の一例を以下に示す．

- **プロンプト A**
  Low Stock: Indicates that a product is limited in quantity, even though that claim is misleading or false.
- **プロンプト B**
  Low Stock: Indicates that a product is limited in quantity to induce urgency.
  - **Target**: "Only 3 left", "Low stock", "Almost gone", "2 people have this in their cart".
  - **EXCLUDE**: "Out of stock", "Sold out", "Not available", "Backorder", "In Stock". (Do NOT detect items that cannot be purchased immediately).

4.3 正解の付与

本実験では，提案手法の評価指標として，適合率（Precision）を評価指標として採用した．現実の Web サイトにおけるダークパターンの網羅的な目視確認（再現率の算出）は非現実的であるため，本研究では「SLM がダークパターンであると警告したものが，実際に真のダークパターンであったか」という，誤検知の少なさに焦点を当てて評価を行う．

正解の付与は以下の手順で実施した．まず，推論パイプラインを通じて出力された JSON 結果のうち，モデルが 1 つ以上のダークパターンを検出したチャンク（陽性予測データ）をすべて抽出する．次に，抽出された要素（HTML タグ）とモデルが提示した推論理由（Reasoning）を，対象ページの実際のレンダリング結果および前後のコンテキストパス（DOM の階層情報）と照らし合わせる．人間の評価者が Gray らの定義に基づいて目視判定を行い，ダークパターンであると確認できた場合を真陽性（True Positive: TP），通常の UI 要素の誤検知や，モデルによる存在しない要素の捏造（ハルシネーション）を偽陽性（False Positive: FP）として分類した．

4.4 実 験 結 果

4.4.1 推論パフォーマンス

構築した推論パイプラインを用いて，200 ページの処理を

表 3　プロンプト設計（A：定義のみ，B：定義＋除外条件）によるカテゴリ別検知結果の比較

	プロンプト A			プロンプト B（除外条件付）		
DP タイプ	TP 数	FP 数	適合率	TP 数	FP 数	適合率
Low Stock	27	59	0.3140	26	18	0.5090
Countdown Timer	1	24	0.0400	1	17	0.0556
Limited Time Messages	2	12	0.1429	2	30	0.0625
High Demand	0	5	0.0000	0	1	0.0000
Parasocial Pressure	0	5	0.0000	0	0	–
Confirmshaming	0	1	0.0000	0	0	–
Endorsements and Testimonials	0	1	0.0000	0	0	–
Activity Messages	0	0	–	0	0	–
架空のタイプ	0	4	0.0000	0	16	0.0000
全体	**30**	**111**	**0.2128**	**29**	**82**	**0.2613**

行った．この際，バッチ数を 2, 4, 8 に変化させて，それぞれの合計処理時間，平均 VRAM 使用量，最大 VRAM 使用量を記録した．この結果を表 2 に示す．この結果より，VRAM16GB 環境において Qwen2.5-coder-7B-Instruct を用いて安定した動作が可能であることが確認できる．

4. 4. 2　プロンプト設計による検知精度の比較

プロンプトの記述粒度が SLM の推論精度に与える影響を評価するため，パターンの定義文のみを与えた「プロンプト A」と，具体例および厳密な除外条件を付与した「プロンプト B」による推論結果の比較を行った．各プロンプトにおける DP タイプ別の真陽性（TP）数，偽陽性（FP）数，および適合率を表 3 に示す．

全体の結果として，プロンプト A を用いた場合の適合率は 0.2128（21.28%）であったのに対し，プロンプト B を用いた場合は 0.2613（26.13%）となり，約 5%の精度向上が確認された．しかし，カテゴリ別の検出内訳を分析すると，プロンプトの詳細化がもたらした効果は一様ではなく，特定のカテゴリにおける著しい精度改善と，別のカテゴリにおける深刻な推論の崩壊という両極端な結果が混在していることが判明した．

a）　除外条件の成功例

プロンプト B における明確な改善効果は，「Low Stock」カテゴリにおいて確認された．プロンプト A では FP が 59 件発生し適合率が 0.3140 であったが，プロンプト B では TP 数を概ね維持したまま FP を 18 件へと大幅に抑制し，適合率を 0.5090 に向上させることに成功した．これは，プロンプト B に記述した「"Out of stock" や "Sold out" は今すぐ購入できないため検知してはならない」という否定形の除外条件をモデルが正しく解釈し，単純なキーワードマッチングによる誤検知を回避できたためであると言える．

b）　複雑なルール付与による推論崩壊（ハルシネーションの増加）

一方で，プロンプトの詳細化は予期せぬ重大な副作用をもたらした．「Limited Time Messages」カテゴリにおいては，プロンプト B を適用したことで FP が 12 件から 30 件へと激増し，適合率が 0.1429 から 0.0625 へと悪化した．さらに致命的な結果として，プロンプトに定義されていない「架空の DP タイプ」を出力するハルシネーション（形式的崩壊）の件数が，プロンプト A の 4 件からプロンプト B では 16 件へと 4 倍に増加した．

これらの結果は，7B パラメータクラスの小規模言語モデルにおいて，Zero-Shot のプロンプトエンジニアリングのみで複雑なタスクを制御しようとするアプローチの限界を定量的に示している．モデルに対して「例示」や「〜してはいけない」という多数の制約を同時に与えた結果，プロンプトのコンテキストが複雑化・長大化し，モデルが情報過多（オーバーフィット）に陥ったと推察される．一部のルール（Low Stock の除外など）には適合できたものの，ルールの全体像を矛盾なく保持し続ける処理能力が不足しており，結果として定義の拡大解釈（Limited Time Messages の誤検知増）や，指示されていない架空のルールを捏造するハルシネーションを引き起こしたと考えられる．

以上の結果から，In-the-wild の無秩序な Web データに対し，ローカル SLM を用いて実用的な精度の静的解析を行うためには，プロンプトの工夫のみに依存することには限界があることが明らかとなった．

5　おわりに

本稿では，日本語 EC サイトにおけるダークパターンの実態解明に向けた第一歩として，SaaS 型 LLM に依存しないローカル SLM を用いたダークパターン検知手法を提案した．具体的には，レンダリングを伴う「視覚情報埋め込みを伴う構造保持型 DOM 分割」アルゴリズムを導入することで，LLM が解釈可能な形式で視覚情報をテキスト化し，これを 4bit 量子化された Qwen2.5-Coder-7B に入力することで，低コストかつ自律的なソースコード解析を実現するフレームワークを構築した．本手法は，API コストや外部サービスの仕様変更といった制約に左右されず，比較的小規模な計算資源（VRAM16GB 程度）で大規模な調査を可能にする点において，学術的および実用的な意義を持つ．

評価実験の結果，詳細な除外条件を付与した Zero-Shot プロンプトにより，特定のダークパターン（希少性の煽り等）にお

いては誤検知の抑制が可能であることが示された．しかし同時に，本研究のアプローチが抱える根本的な課題が浮き彫りとなった．

一つ目の課題は， 単体テストを行わずに検知パイプライン全体でのみ評価したことである．時間の都合上このような評価方法になってしまったが，単体テストを行っていないことにより，どこに問題があったのかなど，詳細な考察が行えなかった．

二つ目の課題は，前処理の方法である．本稿では，WARC ファイルをレンダリング済み HTML に変換し，これを入力としたものの，ダークパターンの判定に必須なものは，WEB ページ上にあるテキスト情報，文字やボタンなどの色や大きさなどといいた視覚的情報である．それに対して，レンダリング済み HTML はその大半が不要なデータであり，前処理方法の再検討が必要である．

三つ目の課題は，評価方法である．本稿では，陽性データに正解を付与することで適合率を算出し評価を行ったものの，ダークパターンの検知においては，再現率を用いた取りこぼしの評価が重要であると考える．しかし，実際の WEB サイトデータに対してダークパターンの取りこぼしなく正解データを付与することは難しい．このことから，検知手法の性能を評価するためには，正解ラベル付きのデータセットを用いることが必要であると考える．

今後の方針として，まずは正解ラベル付きの合成データセットを行う．架空の WEB サイトを作成し，これをデータセットとして保存することにより，適合率，再現率，F 値などの算出を可能にする．

次に，前処理をふくむ検知パイプラインの改良を行う．その上で，ダークパターンの検出実験を行い，課題を探る．

文　献

[1] ハリー・ブリヌル（著）, 高瀬みどり（訳）, “ダークパターン 人を欺くデザインの手口と対策,” 株式会社 BNN, p.12 (2024).

[2] OECD, “Dark commercial patterns,” OECD Digital Economy Papers, No.336, pp.30–45, October 2022.

[3] Colin M. Gray, Cristiana Teixeira Santos, Nataliia Bielova and Thomas Mildner, “An Ontology of Dark Patterns Knowledge: Foundations, Definitions, and a Pathway for Shared Knowledge-Building,” CHI’24: Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems, No.289, pp.1–22 (2024).

[4] Yuki Yada, Jiaying Feng, Tsuneo Matsumoto, Nao Fukushima, Fuyuko Kido, and Hayato Yamana, “Dark patterns in e-commerce: a dataset and its baseline evaluations,” 2022 IEEE International Conference on Big Data, pp.3015–3022 (2022).

[5] S M Hasan Mansur, Sabiha Salma, Damilola Awofisayo and Kevin Moran, “AidUI: Toward Automated Recognition of Dark Patterns in User Interfaces,” ICSE ’23: Proceedings of the 45th International Conference on Software Engineering, pp.1958-1970 (2023).

[6] Zewei Shi, Ruoxi Sun, Jieshan Chen, Jiamou Sun, Minhui Xue, Yansong Gao, Feng Liu and Xingliang Yuan, “50 Shades of Deceptive Patterns: A Unified Taxonomy, Multimodal Detection, and Security Implications,” WWW ’25: Proceedings of the ACM on Web Conference 2025, pp.978–989 (2025).

[7] Binyuan Hui, Jian Yang, et al. "Qwen2.5-coder technical report," arXiv preprint arXiv:2409.12186 (2024).

[8] 久富 望, “ローカル PC での LLM（大規模言語モデル）について, ” 知能と情報，36 巻 3 号，pp.70–76 (2024).