EPiC Series in Computing

Volume 111, 2026, Pages 33–48

Proceedings of The 21st Asia Joint Conference on Information Security

Beyond Classification Accuracy: An Exploration-Range Evaluation of Adaptive Crawling for Fake Shopping Sites

Kotono Karasawa[1], Shingo Matsugaya[123], Makoto Shimamura[2], and Masaki Hashimoto[1]

[1] Graduate School of Science for Creative Emergence, Kagawa University
hashimoto.masaki@kagawa-u.ac.jp
[2] Trend Micro, Inc.
[3] Japan Cybercrime Control Center

Abstract

In recent years, fake shopping sites targeting Japanese users have appeared in the top results of search engines through SEO poisoning, causing increasing damage. Conventional collection methods rely on fixed keywords and cannot keep up with evolving attack campaigns, delaying the discovery of new sites. We propose a closed-loop crawler that incorporates the page-level outputs of a fake-site classifier (fastText+LightGBM) into the search queries of the next cycle. Search queries are generated by a seed-compound strategy that combines characteristic words extracted from positive pages with seed words from the fake-shopping context (e.g., "deep discount," "official"). To complement evaluations that tend to focus on classifier accuracy, we also introduce per-cycle new-host counts and cumulative unique-host counts as exploration-range metrics. In a comparative experiment ($n = 3$ for the proposed method, $n = 2$ for the baseline), the fixed-keyword baseline yielded zero new-host acquisition from cycle 2 onward, indicating complete stagnation, whereas the proposed method continued to discover new hosts and, at cycle 3, achieved a cumulative unique-host count approximately 7.6 times that of the baseline on average.

Keywords: Fake Shopping Sites, SEO Poisoning, Adaptive Crawling, Closed-Loop Crawler, Seed-Compound Strategy, Exploration-Range Evaluation

1 Introduction

In recent years, attacks that redirect users from web search results to fraudulent sites such as fake shopping sites have been continuously observed, and public agencies and security vendors have repeatedly issued warnings[1, 2, 3]. In particular, SEO poisoning is a technique that exploits SEO to make attack-related pages appear in the top results of search engines; it has been reported to redirect users from search queries that combine terms such as "official," "deep discount," and "in stock" with timely topics (e.g., Osaka-Kansai Expo 2025, the government rice stockpile)[2, 3]. Since attack campaigns continuously change in both search terms and redirection destinations, one-off investigations and fixed exploration terms tend to delay the discovery of new sites[4, 5].

In this study, we focus on fake shopping sites targeting Japanese users. Among the sites to which users may be redirected through search results under SEO poisoning, we particularly target these fake shopping sites and the stepping-stone pages (compromised legitimate sites) in their redirection chains[4]. Because these sites can appear in large numbers and have short lifetimes, continuous collection and updating is a prerequisite for analysis and countermeasures. However, conventional collection based on fixed keywords tends to stagnate in finding new sites when search results converge to the same set of sites.

Therefore, the objective of this research is to construct an automatic collection and analysis framework that starts from search results to collect URLs and, by cycling through storage, classification, and exploration-term updates, continues to discover new fake shopping sites while adapting to evolving attacks. The contributions of this work are threefold:

1. **Design of a closed-loop crawler.** We design a closed-loop architecture in which the page-level outputs of a classifier (fastText+LightGBM) are fed back into the set of search queries for the next cycle, aiming at fake shopping site discovery. The search queries are generated by a *seed-compound* strategy that combines characteristic words extracted from positive pages with seed words from the fake-shopping context (e.g., "deep discount" (*gekiyasu*), "official" (*kōshiki*), "in stock" (*zaiko ari*)), maintaining the exploration intent aligned with attack pathways while expanding reach.

2. **Introduction of exploration-range evaluation.** To complement existing evaluations that tend to focus on classifier accuracy, we introduce per-cycle new-host counts (`new_hosts`) and cumulative unique-host counts (`cum_unique_hosts`) as exploration-range metrics. These metrics quantify the ability to "continue discovering new sites" under evolving SEO-poisoning attack campaigns.

3. **Empirical validation.** Through a comparative experiment on Japanese fake shopping sites ($n = 3$ for the proposed method, $n = 2$ for the baseline), we show that the fixed-keyword approach yields zero `new_hosts` from cycle 2 onward, indicating complete stagnation in exploration (cycle 1 also produces only 1.0 hosts on average, which is limited). In contrast, the proposed closed-loop approach continues to acquire new hosts across cycles and, at cycle 3, achieves a cumulative unique-host count approximately 7.6 times that of the fixed-keyword approach on average.

The remainder of this paper is organized as follows. Section 2 reviews related work and clarifies the position of this study. Section 3 describes the architecture and processing flow of the closed-loop crawling system, as well as the automatic keyword generation method. Section 4 presents the experimental setup and evaluation metrics, and Section 5 presents the results and an analysis of generated keywords. Section 6 discusses threats to validity, limitations, and operational considerations. Section 7 concludes the paper and outlines future work.

2 Related Work

Related work can be organized into the realities of fake shopping site attacks and their detection (Section 2.1), the difficulty of observation and collection strategies (Section 2.2), and the position of this study (Section 2.3).

2.1 Attack Realities and Detection of Fake Shopping Sites

Fake shopping site damages have been continuously reported, with topics and contexts of redirection changing depending on the situation[2, 3, 1]. Kodera et al. analyzed site structure and operational characteristics[6], while Michishita et al. reported cases of compromised legitimate sites being exploited as stepping stones from search results to fake shopping sites[4]; both fake shopping sites (final destinations) and stepping-stone pages (entry points) should therefore be included as targets for collection. A collection method for fake shopping sites that reuse product information from legitimate sites has also been proposed[7].

For detection, the field has evolved from learning-based detection using lightweight URL features[8] to deep-learning-based URL representation learning[9]; for Japanese fake shopping sites, a system combining fastText and LightGBM has been proposed[10]. Other approaches include visual features[11], LLMs with knowledge graphs[12], and large-scale fraudulent e-commerce site detection[13, 14]. As detectors become more sophisticated, the continuity of data collection that underpins evaluation and retraining becomes equally important.

2.2 Observation Challenges and Collection Strategies

Search-engine abuse (SEO poisoning) has long been observed as search-result contamination and redirection attacks[5, 15], where changes in search rankings and redirection destinations can cause stagnation in fixed-keyword collection. Crawlers also face evasion such as cloaking[16, 17], and in the phishing field, blacklist evasion measurement[18], large-scale cloaking detection[19], and responses to CAPTCHA-based serving[20] have been reported, suggesting that collection strategies should be evaluated by reach range and stagnation in addition to detection accuracy.

To address these challenges, guided and adaptive collection has been studied[21], and large-scale detection of defacements associated with search-engine abuse campaigns has also been reported[22]. To keep up with adversarial changes, mechanisms that continuously update exploration terms are key, which motivates the closed-loop approach pursued in this study.

2.3 Position of This Study

Existing studies on detection models[10, 9, 11] primarily aim at improving classification accuracy, and the design of collection strategies themselves is not their main concern. Studies addressing collection strategies[21, 22] target guided collection or large-scale campaign detection, but a closed-loop structure that directly feeds back detection results into next-cycle search-query generation is not necessarily foregrounded. An existing method for collecting fake shopping sites[7] uses popular product information as search queries but lacks an automatic query-update mechanism based on collection results. In contrast, this study proposes and evaluates a closed-loop crawler that feeds detection results back into search-query updates, demonstrating effectiveness through a relative comparison via the exploration range (`cum unique hosts`) and the stagnation metric (`new hosts`).

3 System Design

3.1 Closed-Loop Architecture

The proposed system adopts a closed-loop architecture in which classification results are fed back into the search-query generation of the next cycle. It consists of five modules: (1) a search and collection module, (2) an SQLite database, (3) a feature extraction module, (4) a

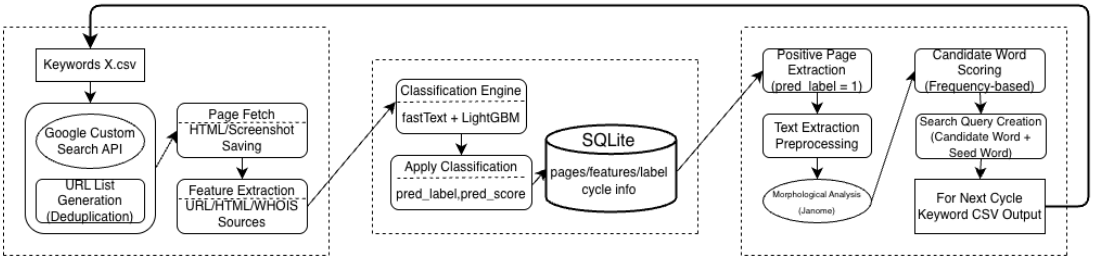

Figure 1: Overall system architecture: a closed-loop of collection → classification → search-query update. The page-level outputs of the classifier (fastText+LightGBM) are fed back into the search-query generation of the next cycle.

classification engine (fastText+LightGBM), and (5) an automatic keyword generation module (Figure 1). The feedback of classification results (Module 4 → 5 → 1) is the core of this system; below, we describe the design of each module.

Each cycle is executed in the following steps:

1. **Search and collection.** Obtain URLs via the search API and save HTML through HTTP fetching (with browser-based fetching as needed).

2. **Classification.** Extract features from the saved HTML, and assign `pred_label` (positive/negative) and `pred_score` using the trained classifier.

3. **Search-query generation (Condition B only).** Extract candidate words from the set of pages with `pred_label`=1, and generate a keyword CSV for the next cycle.

4. **Storage and aggregation.** Save the database for each cycle and aggregate metrics (unique hosts, etc.).

Collected data are centrally managed in SQLite. The URL, fetch time, cycle number, HTML save path, screenshot save path, `pred_label`, and `pred_score` are stored, enabling cycle-wise comparison and re-analysis. WHOIS information is unstable in retrieval and prone to missing values; we did not fully leverage it in this experiment (discussed in Section 6.4).

3.2 Search and Collection Module

This module is responsible for sending queries to the search API and fetching the HTML of the resulting URLs. In what follows, we describe the search-result retrieval configuration, URL normalization, and handling of fetch failures and cloaking, in this order.

Search-Result Retrieval Configuration. We use the Google Custom Search API[23] to retrieve search results. Due to API limits on the number of results per call, the implementation is configured to retrieve 3 results per API call and 2 pages per keyword, so that the theoretical maximum is 6 URLs per keyword (the actual number is less due to deduplication and fetch failures).

Table 1: Examples of features (representative implementation).

Type	Examples (overview)
URL	URL length, domain length, digit/symbol ratio, TLD type.
HTML	Extracted text length, word count, link count (simple), fastText representation, etc.
WHOIS	Domain age, registration country, registrar (when available), etc.

URL Normalization and Deduplication. The search API may return the same page through different queries, and may also return many URLs that differ only in parameters within the same host. From the perspectives of exploration-range evaluation (at the host level) and collection cost, we therefore designed the system to suppress duplication before and after fetching. Specifically, we introduced (i) URL normalization (scheme unification, trailing-slash unification, removal of known tracking parameters), (ii) URL-level deduplication, and (iii) a per-host upper bound (a cap on the number of fetches from the same host), preventing a single cycle from being disproportionately biased toward the same target. However, overly strong normalization can cause us to miss distinct pages; since this paper's main objective is to compare exploration ranges, we use host novelty (`new_hosts`) as the central metric and leave URL-level coverage for future work.

Handling of Fetch Failures and Cloaking. Web fetching may yield unstable HTML due to access denial, timeouts, dynamic rendering, or content differences depending on the access source (cloaking). In our implementation, HTTP fetch failures are retried; if they still fail, the reason for failure (HTTP status, exception type) is recorded in the database. We can also switch to a headless browser to save HTML and screenshots, which can be used for visual inspection and downstream analysis. However, since our evaluation is limited to successfully fetched pages, the impact of fetch failures on exploration-range metrics (e.g., bias caused when certain host groups are difficult to fetch) remains. This issue is discussed as a threat to validity in Section 6.4.

3.3 Feature Extraction and Classification Engine

For classification, we use URL information, HTML text information, and WHOIS information (when available). Table 1 shows examples of features. In addition to simple features such as URL length and the ratio of special characters, surface-level features extracted from HTML (e.g., word count) and fastText embeddings are fed into LightGBM.

The classifier follows the framework of prior work: it converts the body text extracted from HTML into a distributed representation (document vector) using fastText, and performs binary classification with LightGBM, combining the URL/HTML additional features[10]. **Our claim does not rely on high classifier accuracy.** Since the goal of this research is to evaluate an exploration framework that continues to discover new sites, neither novel proposal of the classifier itself nor its rigorous comparison is within our scope. In the evaluation, we use the same classifier and the same settings for both conditions A and B, and discuss the relative comparison in terms of exploration-range and stagnation metrics.

For reference, the classifier's accuracy on an evaluation set of 42 instances is: Accuracy=0.929, Positive Recall=1.000, Negative Recall=0.625, indicating that false positives may remain (these are reference values for understanding classifier characteristics and do not constitute the basis of this paper's claims). Therefore, our `pos` counts do not guarantee the true number of fake sites. Nonetheless, since the exploration feedback is generated from "the posi-

Table 2: Examples of candidate-word filtering (representative implementation).

Item	Example rules
Length	Exclude words of length 1; symbol-only or digit-only words
Part of speech	Primarily nouns (exclude symbols, particles, auxiliary verbs)
Normalization	Unify full-width/half-width forms; unify upper/lower case (alphabet)
Stop words	Template words (e.g., login," cart")

tively predicted set," classifier errors can affect keyword quality (Section 6.3).

3.4 Automatic Keyword Generation

The automatic keyword generation module takes the set of pages with `pred_label`=1 as input, and performs preprocessing (HTML → text), morphological analysis, candidate-word extraction, scoring, query generation, and CSV output. For morphological analysis, we use Janome[24]; we adopt nouns primarily and exclude single-character words, symbol-only words, digit-only words, etc. to form the candidate-word set. For scoring, to ensure stable operation in small-scale experiments, we use a frequency-based scheme and use the top K words ($K = 20$) as the core of the next-cycle exploration.

Candidate-Word Filter. The noun sequence extracted by morphological analysis often contains template words and generic e-commerce terms. We therefore introduce a simple rule-based filter that is stable even in small-scale experiments, to suppress the inclusion of overly generic or noisy words. Table 2 shows examples of candidate-word filtering. Because filters need updating as exploration targets evolve, we implement the current set as fixed provisional rules in this paper, and discuss improvements in Section 6.3.

Seed-Compound Strategy. If we use only candidate words extracted and filtered from positive pages as search queries, the inclusion of generic words can cause the search scope to diffuse and drift away from the goal (fake shopping site discovery). To address this, we generate queries that combine a candidate word w derived from a positive page with a seed word s from the fake-shopping context (e.g., "deep discount," "in stock," "official, " "ōdoriyasu," "mail order"). We refer to this as the **seed-compound strategy**. These seed words were selected based on alerts and investigation reports indicating that fake shopping sites convey trustworthiness and a sense of bargain by emphasizing "extremely low prices below market rate" and "official/authentic" to encourage purchases[2, 3, 1].

Algorithm. The generation process is organized as the following algorithm. Let S_c be the set of positive pages at cycle c, $T(p)$ the text extracted from page p, and $Tok(\cdot)$ the morphological analysis result (noun sequence).

1. For each page p in S_c, extract $T(p)$, and obtain a candidate-word set W from $Tok(T(p))$.
2. Apply the rules in Table 2 to W, and aggregate the occurrence count $f(w)$ of each word.
3. Select the top K words by $f(w)$ ($K = 20$), and combine them with the seed word set $Seed$ to generate queries Q (e.g., $w+$ "deep discount").
4. Output Q as a CSV and use it as the search input for the next cycle.

The frequency-based scheme is easy to implement and works at small scale, but it carries the risks of generic-word contamination and lack of diversity; subsequent improvements (e.g., TF-IDF, phrase extraction, score weighting) are discussed in Section 6.3.

4 Evaluation

4.1 Comparison Conditions

Condition A (A fixed) explores using six initial keywords fixed throughout cycles 0–3. The initial keywords (used in cycle 0) were chosen to cover fake e-commerce, tech-support scams, and warning scams: three product-related ones (`PS5 in stock`, `iPhone deep discount`, `Louis Vuitton authentic cheap`), two tech-support ones (`Apple ID locked`, `Amazon inquiry`), and one warning one (`virus warning fake`), totaling six. Condition B (B boost) uses the same initial keywords as Condition A at cycle 0; from cycle 1 onward, it updates the search using the top 20 keywords automatically generated by the seed-compound strategy described in Section 3.4. Here, `boost` refers to the expansion of exploration range (exploration boost) and is unrelated to machine learning methods such as gradient boosting.

Note that the initial keywords also include phrases from attack contexts other than fake shopping (i.e., tech-support and warning scams). This is intended to confirm a behavior in which, even when exploration begins from diverse entry points, the queries in subsequent cycles converge to the fake-shopping context through the feedback via the classifier (Japanese fake shopping site classification) in the closed loop. In fact, all generated keywords from cycle 1 onward, shown in Table 6, are composed in combination with seed words from the fake-shopping context, confirming the natural narrowing function of the closed loop.

The two conditions differ in the number of queries per cycle: Condition A always issues the same six fixed keywords, whereas Condition B issues up to $K = 20$ generated keywords from cycle 1 onward. This asymmetry is intentional rather than a confounding factor, since the inability to expand the query set in response to collected results is itself the central limitation of fixed-keyword collection that this study aims to expose; indeed, merely increasing the number of *fixed* keywords would not resolve the stagnation, as a fixed set converges to the same host cluster regardless of budget (Condition A reaches `new hosts`=0 from cycle 2 onward, Section 5). A strictly budget-matched baseline that separates the effects of query *quantity* and query *renewal* is left for future work.

Note also that, since the results of the search API can vary depending on time, both conditions are independently executed multiple times to evaluate them while taking variability into account. The details of independent runs are described in Section 4.2.

4.2 Variability Evaluation through Independent Runs

Since the results of the search API can vary depending on time and date, in this study we independently ran each condition multiple times as follows. Condition B (B boost) was run three times (B-1, B-2, B-3) and Condition A (A fixed) was run twice (A-1, A-2), independently. Each independent execution (Run) shared the same full configuration—including the initial keyword set, seed-word set, classifier (the trained model of fastText+LightGBM), search-API parameters (region, language, number of results to retrieve, etc.), and the number of cycles (three)—and varied only the **start date and time of execution**. This setting allows us to evaluate both the reproducibility of the closed loop as a whole under the same configuration and the impact of time-dependent variations on the search API side.

In Section 5, we mainly report the cross-Run mean and standard deviation of the main metrics (`cum_unique_hosts`, `new_hosts`) for each condition. Per-Run values are reported alongside in the appendix or within tables in the main text.

In this study, Condition B was run three times and Condition A was run twice. Although the sample size is limited, the main claim of this study concerns the comparison of exploration-range metrics between Conditions B and A, and the effect size (B/A ratio) of approximately 7.6 times is so large that the difference between the conditions can be clearly separated even with a limited sample size. Specifically, even in the worst pairwise comparison between individual Runs (B-2/A-1 = 6.7 times), Condition A is exceeded by more than five times, and the best comparison (B-3/A-2 = 8.3 times) shows essentially the same trend (Section 5). Therefore, although additional experiments would be required to perform stricter statistical tests (e.g., a t-test), within the experimental scale of this study, we judge that the possibility of the difference between conditions arising by chance is essentially excluded.

4.3 Collection Pipeline and Settings

Pipeline. In this study, we define one cycle as "search → fetch → store → classify → search-query update." Among the processes in each cycle, the factors that directly affect the exploration range (the number of reached hosts) include: (i) bias in URLs at the top of search results, (ii) URL duplication, (iii) fetch failures (denials, timeouts, etc.), and (iv) collection of multiple pages from the same host. We therefore apply the following preprocessing policies:

- **URL-level deduplication.** The same URL is stored and classified only once.
- **Host extraction.** The host part is extracted, and exploration-range metrics are aggregated at the host level (`unique hosts`).
- **Definition of fetch success.** An item is counted as `total` if its HTML could be saved through HTTP fetching (with browser-based fetching as needed).
- **Storage.** The HTML path, screenshot path, fetch time, cycle number, and classification result are recorded in SQLite.

Classifier. As described in Section 3.3, the classifier follows the framework of prior work[10], and **our claim does not rely on high classifier accuracy**. We use the same classifier and the same settings for both Conditions A and B, and discuss the **relative comparison in terms of exploration-range and stagnation metrics**. Since our `pos` is a predicted label rather than the ground truth, classifier errors can affect keyword quality (Section 6.3).

Keyword generation. For Condition B, we apply the seed-compound strategy in Section 3.4 with $K = 20$, using the set of positive pages (pred label=1) from the previous cycle as input. The seed-word set is the same as in Section 3.4.

4.4 Evaluation Metrics

We evaluate the exploration range using the following metrics. Here, the host is extracted from the URL's host part, and uniqueness is determined at the host level.

- `total`: the number of items that were fetched, saved, and became targets of classification (not the total hit count of the search engine).

- `pos`/`neg`, `pos rate`: the number and ratio of items predicted as positive (fake candidates) and negative by the classifier.
- `unique hosts`: the number of unique hosts observed in the cycle.
- `new hosts`: the number of new hosts not observed before the current cycle (important for stagnation detection).
- `cum unique hosts`: the cumulative number of unique hosts (the main metric of reach range).

To assess the "efficiency" of exploration, we also use derived metrics: `new hosts` / `total` (new-reach rate) and `unique hosts` − `new hosts` (the number of revisited known hosts; degree of duplication).

5 Results

5.1 Per-Cycle Aggregation

Table 3 shows the per-cycle aggregation results (mean±SD), and Table 4 shows the per-Run values of `cum unique hosts`. The aggregation is based on independent runs (Section 4.2) with $n = 2$ for Condition A (A fixed) and $n = 3$ for Condition B (B boost).

Condition A (A fixed) observed an average of `unique hosts`=29.0±2.8 at cycle 0; from cycle 2 onward, `new hosts` became 0, with search results converging to the same set of sites and exploration completely stagnating. Consequently, `cum unique hosts` also remained at 30.0±1.4 from cycle 2 onward. At cycle 1, a small acquisition of `new hosts`=1.0±1.4 was observed, but this is attributable to slight time-dependent fluctuations in the search API results; the trend of early-stage stagnation, with cycle 2 onward becoming completely 0, is common to both runs (Table 4).

In contrast, Condition B (B boost) continued to reach new regions, with `new hosts`=85.3±7.6 at cycle 1 and `new hosts`=87.3±5.5 at cycle 2; `cum unique hosts` reached 229.0±18.4 at the end of cycle 3. Across all three runs, continuous acquisition of new hosts from cycle 1 to cycle 3 was confirmed (B-1: 237, B-2: 208, B-3: 242), indicating reproducibility under the same configuration. Although `new hosts` at cycle 3 decelerated to 29.0±4.4 compared with cycles 1 and 2, continuous discovery of new hosts was nevertheless maintained.

In terms of the mean comparison of `cum unique hosts`, Condition B reached approximately **7.6 times** that of Condition A (229.0/30.0). This indicates that updating search queries based on classifier outputs can avoid the stagnation (`new hosts`=0) that tends to occur in search-based collection and incrementally expand the range of reachable hosts, as demonstrated reproducibly across multiple independent runs.

Furthermore, while `pos rate` stayed around 0.20–0.23 for Condition A, Condition B maintained higher levels: 0.34±0.04 at cycle 1, 0.38±0.08 at cycle 2, and 0.33±0.05 at cycle 3. Although our `pos` is a predicted label rather than the ground truth, as a relative comparison under the same classifier and the same settings, this suggests that search-query updates not only expand the reach but also reach regions where positive predictions are more likely to be obtained. Note that `pos rate` showed some variability across runs (especially pronounced in Condition A), confirming that even under the same configuration, time-dependent fluctuations in the search API can affect the proportion of pages predicted as positive.

Table 3: Per-cycle aggregation results for each condition (mean±SD; Condition A: $n = 2$, Condition B: $n = 3$).

Condition	cycle	total	pos rate	unique hosts	new hosts	cum unique hosts
A fixed (n=2)	0	35.5±0.7	0.20±0.12	29.0±2.8	29.0±2.8	29.0±2.8
A fixed (n=2)	1	35.5±0.7	0.23±0.12	29.0±2.8	1.0±1.4	30.0±1.4
A fixed (n=2)	2	35.5±0.7	0.21±0.10	29.0±2.8	0±0	30.0±1.4
A fixed (n=2)	3	35.5±0.7	0.23±0.12	29.5±2.1	0±0	30.0±1.4
B boost (n=3)	0	35.0±1.7	0.22±0.10	27.3±3.2	27.3±3.2	27.3±3.2
B boost (n=3)	1	109.7±3.8	0.34±0.04	90.3±6.0	85.3±7.6	112.7±9.2
B boost (n=3)	2	115.0±3.6	0.38±0.08	98.0±6.1	87.3±5.5	200.0±14.2
B boost (n=3)	3	39.3±2.9	0.33±0.05	37.3±1.2	29.0±4.4	229.0±18.4

Table 4: Per-Run values of `cum unique hosts`.

Run	cycle 0	cycle 1	cycle 2	cycle 3
A-1	31	31	31	31
A-2	27	29	29	29
B-1	31	118	205	237
B-2	25	102	184	208
B-3	26	118	211	242

5.2 Derived Metrics

Table 5 summarizes the derived metrics defined in Section 4: new-reach rate (`new hosts` / `total`) and known revisits (`unique hosts` − `new hosts`).

For Condition A, from cycle 2 onward, the new-reach rate is 0 and known revisits remain at about 29, indicating that the search results are completely fixed within the known set. The new-reach rate at cycle 1 is 0.03±0.04, which is extremely low, indicating that stagnation is established early. In contrast, for Condition B, the new-reach rate remains at high levels of 0.74–0.78 throughout cycles 1–3, demonstrating that the search-query updates strongly contribute to new-host reaching. Known revisits transition to 10.7±3.2 at cycle 2 and 8.3±5.1 at cycle 3, with a particular increase at cycle 2. This suggests that some keywords may have returned to known clusters, which is related to keyword quality (genericization and templatization); this is discussed in Section 6.3.

5.3 Generated Keywords across Cycles

Table 6 shows examples of keywords generated under Condition B (the top 3 actually used for search in each cycle, for all 3 Runs). The lexical composition varied substantially across Runs, reflecting time-dependent variations in search-API results that cause the initial positively predicted page set to differ across Runs. Nonetheless, across all Runs, the composite structure of "candidate word + seed word" from the fake-shopping context ("deep discount," "in stock," "authentic," etc.) was consistently preserved, demonstrating that the seed-compound strategy functions robustly against variations in initial conditions and underlies the stable exploration-range expansion observed in Table 3.

On the other hand, across all Runs, generic e-commerce terms and common words such as "favorites," "shipping fee," "top," "case," and "order" appeared in the top positions. While such words contribute to expanding the exploration range, even when combined with seed words they do not necessarily improve redirection efficiency to fake sites, and tend to return to known clusters. The fact that, according to the derived metrics (Table 5), known revisits show an

Table 5: Derived metrics (new-reach rate and number of known revisits, mean±SD).

Condition	cycle	new hosts / total	Known revisits (unique - new)
A fixed (n=2)	0	0.82±0.06	0±0
A fixed (n=2)	1	0.03±0.04	28.0±4.2
A fixed (n=2)	2	0±0	29.0±2.8
A fixed (n=2)	3	0±0	29.5±2.1
B boost (n=3)	0	0.78±0.07	0±0
B boost (n=3)	1	0.78±0.06	5.0±1.7
B boost (n=3)	2	0.76±0.06	10.7±3.2
B boost (n=3)	3	0.74±0.15	8.3±5.1

Table 6: Examples of generated keywords under Condition B (top 3 per cycle, 3 Runs).

cycle	B-1	B-2	B-3
1	Casino deep discount / Casino in stock / Casino authentic	iPhone deep discount / iPhone in stock / iPhone authentic	shipping deep discount / shipping in stock / shipping authentic
2	favorites deep discount / shipping deep discount / shipping in stock	"-iri" deep discount / size deep discount / shipping deep discount	case deep discount / order deep discount / top deep discount
3	top deep discount / supplies deep discount / top in stock	case deep discount / iPhone deep discount / ballpoint pen deep discount	printing deep discount / signboard deep discount / fee deep discount

increasing trend from cycle 2 onward (cycle 2: 10.7±3.2, cycle 3: 8.3±5.1) is consistent with this. Therefore, jointly achieving exploration-range expansion and maintenance of the positive rate requires scoring and filter design that suppresses the generality of the generated words.

5.4 Summary

From the above, the fixed-keyword approach (Condition A) stagnates early due to the convergence of search results: `cum unique hosts` remained at 30.0±1.4 at cycle 3, as confirmed across 2 Runs. In contrast, the keyword-expansion approach (Condition B) continuously acquired `new hosts` across multiple cycles, and expanded `cum unique hosts` to 229.0±18.4 at cycle 3 (confirmed across 3 Runs). This corresponds to approximately **7.6 times** the value of the fixed-keyword approach, demonstrating that search-query updates contribute strongly to exploration-range expansion, reproducibly across multiple independent runs. Furthermore, since `pos rate` remained at relatively high levels, search-query updates may also contribute to reaching regions that include positively predicted pages.

To confirm the robustness of this value of approximately 7.6 times, we compared `cum unique hosts` across the individual Runs of Condition B's 3 Runs and Condition A's 2 Runs (Table 7). Across all six pairwise comparisons, the ratio ranged from 6.7 times (B-2/A-1) at the minimum to 8.3 times (B-3/A-2) at the maximum, falling within the range of **6.7–8.3 times**. Therefore, the effect of the proposed method exceeding the fixed-keyword approach is stably observed regardless of which individual Runs are chosen.

However, both `total` and `pos rate` varied across cycles and across Runs, suggesting that they may be affected by multiple factors such as search-ranking changes, query genericization, URL duplication and fetch failures, and API constraints. Therefore, when evaluating an exploration strategy, it is important to base the comparison not on `total` alone, but primarily on stagnation metrics (`new hosts`) and reach-range metrics (`cum unique hosts`).

Table 7: Pairwise ratios of `cum unique hosts` between individual Runs (cycle 3, all six pairs).

	B-1 (=237)	B-2 (=208)	B-3 (=242)
A-1 (=31)	7.6	6.7	7.8
A-2 (=29)	8.2	7.2	8.3

6 Discussion

6.1 Stagnation Avoidance and Exploration-Range Expansion

The fact that `new hosts` became 0 from cycle 2 onward in Condition A suggests a situation in which the search results repeatedly converged to the same set of sites, making it difficult to expand into a new exploration space. The `new hosts` at cycle 1 is also limited at 1.0±1.4 on average, indicating that stagnation is established early. While search-based collection is powerful, when the top of the search rankings is fixed to the same cluster, the exploration quickly reaches a plateau[5, 4].

In contrast, in Condition B, search-query updates based on classification results enabled `new hosts` to continue, and `cum unique hosts` increased to an average of 229.0±18.4. Continuous acquisition of new hosts from cycle 1 to cycle 3 was observed across all three Runs (Table 4), and the effect of exploration-range expansion is reproducibly established across independent runs under the same configuration. Since the new-reach rate in Table 5 is high (0.74–0.78 in cycles 1–3), it can be interpreted that the search-query updates worked not by increasing the number of collected pages, but by raising the probability of reaching new hosts.

However, at cycle 3, `new hosts` decelerated to 29.0±4.4 on average, clearly slower than cycle 1 (85.3±7.6) and cycle 2 (87.3±5.5), and this trend was common across all three Runs. This indicates that the closed-loop exploration expansion does not continue indefinitely; in the current implementation of the seed-compound strategy, due to the limits of keyword quality (mixing of generic words), exploration may gradually return to known clusters (discussed in detail in Section 6.3). Therefore, the exploration-range expansion effect of this implementation is clearly established at least at the cycle-3 stage, while keyword quality improvement is a challenge for longer-term continuity.

6.2 Factors of Total Variation and Operational Implications

In Condition B, `total` varied across cycles. In particular, at cycle 3, `total` was clearly smaller than at other cycles across all three Runs (cycle 1: 109.7±3.8, cycle 2: 115.0±3.6, vs. cycle 3: 39.3±2.9), and it is likely that practical operational factors such as constraints on the number of usable search keywords and API constraints overlapped.

The fact that `total` at cycle 3 dropped to about 1/3 of cycles 1 and 2 across all three Runs suggests that this is not an accidental operational constraint of a single Run, but that structural factors inherent to the closed loop itself may also contribute. Specifically, conceivable factors include a decrease in the number of effective candidate words extracted from the positive page set of cycle 2 (which serves as the generation source of cycle 3 search queries), an increase in the duplication of candidate words, and the return to known clusters causing the diversity of the keyword set to decrease. This corresponds to the room for keyword quality improvement described in Section 6.3. However, within the experimental scale of this study, it is difficult to quantitatively separate the structural and operational factors, and disentangling them in larger-scale experiments is a challenge for future work.

The deceleration tendency of `new hosts` also shows similar structurality. In Condition B, the transition of `new hosts` from cycle 1 → cycle 2 → cycle 3 is 85.3 → 87.3 → 29.0 on average, maintaining a high level up to cycle 2 and decreasing to about 1/3 at cycle 3 (the cycle 3 / cycle 1 ratio is approximately 0.34). This transition pattern is common across all three Runs (B-1: 87 → 87 → 32, B-2: 77 → 82 → 24, B-3: 92 → 93 → 31), indicating that the inter-cycle attenuation has structural causes attributable to the keyword generation process on the implementation side. Specifically, in the initial cycles (cycles 1 and 2) of the closed loop, diverse candidate words are extracted from diverse positive page sets, whereas in subsequent cycles, it becomes more difficult to extract new candidate words from the existing positive host set; we refer to this as "local saturation of the exploration space." Therefore, while the current closed-loop implementation shows a clear expansion effect at least within three cycles, for longer-term continuity, mechanisms that suppress the depletion of new candidate words (e.g., the contrast-set introduction and diversity constraints in Section 6.3) become important.

In inter-cycle comparisons, instead of using `total` alone, it is necessary to interpret the transitions of `new hosts` and `cum unique hosts` as primary indicators. This is because, in addition to search APIs only being able to retrieve a portion of the top results, the following factors overlap: (i) URL duplication across multiple keywords, (ii) fetch failures such as access denials and timeouts, (iii) URL-level deduplication, and (iv) API rate limits[23]. In exploration-range evaluation, `new hosts` / `cum unique hosts` should be emphasized over `total` alone. Furthermore, on the operational side, designs that balance cost and observability, such as retrying fetch failures and selecting browser-based fetching (rendering only suspicious pages), are important[18, 19].

6.3 Improving Keyword Quality

While the frequency-based scheme in Section 4.3 is easy to implement and works at small scale, it tends to bring template words and generic e-commerce terms to the top. In this situation, the search terms become generic; while the diversity of search results may increase, the positive rate may decrease. In our experiments, the `pos rate` remained at a stable level, but known revisits (`unique` − `new`) show an increasing trend across all three Runs from cycle 2 onward (cycle 2: 10.7±3.2, cycle 3: 8.3±5.1, Table 5), suggesting that some queries may have returned to known clusters. This is consistent with the structural factors of `new hosts` deceleration at cycle 3 described in Section 6.1 and the `total` decrease described in Section 6.2, supporting the need for keyword quality improvement.

Promising improvements include introducing a contrast set using pred label=0 to compute TF-IDF or log-likelihood ratio for prioritizing words characteristic of the positive side, phrase formation via noun concatenations (2-gram/3-gram) to capture specific named entities rather than generic terms, score weighting based on pred score to prioritize words from high-confidence pages, and diversity constraints (e.g., MMR) to avoid similar words clustering at the top. These are expected to contribute to jointly achieving exploration-range expansion and maintenance of the positive rate.

6.4 Threats to Validity and Limitations

Validity threats. Since this experiment relies on the search API, it can be affected by search-ranking changes, region/language settings, and personalization. For the evaluation of inter-Run variability, we addressed this by independently running Condition B for 3 Runs and Condition A for 2 Runs under the same configuration with fixed search settings (region, language, period; UA, rendering), and reporting the main metrics as mean±SD (Section 4.2). As a result,

while `cum unique hosts` shows some variability at 229.0±18.4 in Condition B, the difference from Condition A (30.0±1.4) is clearly maintained, and the main message (exploration-range expansion by the proposed method) is reproduced across multiple trials.

Since `pos` is a predicted label rather than the ground truth, the discussion of exploration expansion is positioned as "a relative comparison under the same classifier and the same settings." Cross-checking with the ground truth (sample precision evaluation through manual verification) is left for future work.

We provide an additional remark on sample size. Condition B with 3 Runs and Condition A with 2 Runs are limited, and stricter statistical verification through larger-scale independent runs is a challenge for future work. However, the main claim of this study, namely "exploration-range expansion through search-query updates," is based on a large effect size of approximately 7.6 times; even in the worst-case pairwise comparison between individual Runs (B-2/A-1 = 6.7 times), Condition A is still substantially exceeded. Therefore, within the experimental scale of this study, the limitation of sample size is unlikely to overturn the conclusion.

Cloaking and dynamic pages. For stepping-stone sites and fake e-commerce sites, content can change depending on access conditions through cloaking, and the actual nature of pages can be hard to read from static HTML due to dynamic generation[4, 19, 17]. In this study, we perform browser-based fetching and screenshot saving as needed, but full browser rendering is costly. Future challenges include (i) detecting differences between HTTP fetch results and rendering results, (ii) detecting signals such as CAPTCHA, and (iii) interactive fetching only for suspicious pages, to improve observability[20].

WHOIS information. In our experiments, WHOIS retrieval frequently failed, and analysis based on WHOIS information could not be sufficiently performed. WHOIS information may be useful for tasks such as extracting short-lived domains and characterizing biases in registration country and registrar; we leave for future work improving the retrieval success rate (through retry, caching, and TLD-specific handling) and designing features and metrics that assume missing values.

7 Conclusion

In this work, we proposed and evaluated a closed-loop crawling framework for fake shopping site discovery, in which the page-level outputs of a classifier (fastText+LightGBM) are fed back into the search queries of the next cycle through a seed-compound strategy, and introduced per-cycle new-host counts and cumulative unique-host counts as exploration-range metrics. In a comparative experiment ($n = 3$ for the proposed method, $n = 2$ for the baseline), the fixed-keyword approach yielded zero new-host acquisition from cycle 2 onward, while the proposed method achieved a cumulative unique-host count approximately 7.6 times that of the baseline at cycle 3, reproducibly observed across multiple independent runs.

The multi-run evaluation revealed two new insights: the seed-compound strategy maintained the "candidate word + seed word" structure across runs with stable exploration-range expansion of 6.7–8.3 times, demonstrating robustness against initial-condition variations; and a common deceleration of `new_hosts` at cycle 3, which we characterize as "local saturation of the exploration space," indicates that closed-loop expansion is limited to approximately three cycles in the current implementation. The contributions lie along two theoretical axes: the closed-loop architecture, which complements detection-model-focused studies by feeding detection results back into next-cycle query generation, and the exploration-range evaluation, which

complements accuracy-based assessments by quantifying "the ability to continue discovering new sites."

Remaining challenges include validating the classifier through sample-precision evaluation and robustness to distribution shift; improving keyword generation quality via TF-IDF or log-likelihood ratio, n-gram phrase generation, and MMR-based diversity assurance to address "local saturation"; improving robustness against cloaking and dynamic pages; and stable WHOIS retrieval for short-lived domain extraction and registrar/country bias characterization.

Acknowledgments

We acknowledge the use of Claude AI assistant (Anthropic) for the English translation of this paper. This work was supported in part by the JSPS/MEXT KAKENHI under Grant 24K14956.

References

[1] Japan Cybercrime Control Center. Statistical information on malicious shopping sites (first half of 2025). Web. (in Japanese), Accessed: 2025-12-10. https://www.jc3.or.jp/threats/topics/article-641.html.

[2] Trend Micro. Fake shopping sites: Confirmed redirections from search results for osaka-kansai expo goods. Web. (in Japanese), Accessed: 2026-01-10. https://www.trendmicro.com/ja_jp/research/25/i/fake-shopping-sites.html.

[3] Consumer Affairs Agency, Government of Japan. Warning on fraudulent sites disguised as selling rice at low prices. Web. (in Japanese), Accessed: 2026-01-10. https://www.caa.go.jp/notice/entry/043659/.

[4] Daigo Michishita, Satoru Kobayashi, and Toshihiro Yamauchi. Field survey on detecting stepping-stone sites that redirect users to fake shopping sites. In *Proceedings of Computer Security Symposium 2024 (CSS2024)*, pages 1095–1101. Information Processing Society of Japan, Oct 2024. (in Japanese).

[5] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. A nearly four-year longitudinal study of search-engine poisoning. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, CCS '14, page 930–941, New York, NY, USA, 2014. Association for Computing Machinery.

[6] Hirokazu Kodera, Shun Koide, Daiki Chiba, Kazufumi Aoki, and Mitsuaki Akiyama. Elucidating attack methods of fake shopping sites. *IPSJ Journal*, 62:1523–1535, Sep 2021. (in Japanese).

[7] Mitsuho Hasegawa, Kosuke Sekido, Kazuki Takada, Akira Fujita, Rui Tanabe, and Katsunari Yoshioka. Proposal of a collection method for fake shopping sites that reuse product information from legitimate sites. *Proceedings of Computer Security Symposium 2024 (CSS2024)*, pages 1080–1087, 10 2024. (in Japanese).

[8] Justin Ma, Lawrence K. Saul, Stefan Savage, and Geoffrey M. Voelker. Beyond blacklists: learning to detect malicious web sites from suspicious urls. KDD '09, page 1245–1254, New York, NY, USA, 2009. Association for Computing Machinery.

[9] Hung Le, Quang Pham, Doyen Sahoo, and Steven C. H. Hoi. Urlnet: Learning a url representation with deep learning for malicious url detection, 2018.

[10] Keisuke Sakai, Kosuke Takeshige, Kazuki Kato, Naoki Kurihara, Katsumi Ono, and Masaki Hashimoto. An automatic detection system for fake japanese shopping sites using fasttext and lightgbm. *IEEE Access*, 11:111389–111401, 2023.

[11] Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. Phishpedia: A hybrid deep learning based approach to visually

identify phishing webpages. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3793–3810. USENIX Association, August 2021.

[12] Yuexin Li, Chengyu Huang, Shumin Deng, Mei Lin Lock, Tri Cao, Nay Oo, Hoon Wei Lim, and Bryan Hooi. Knowphish: Large language models meet multimodal knowledge graphs for enhancing reference-based phishing detection, 2024.

[13] Marzieh Bitaab, Haehyun Cho, Adam Oest, Zhuoer Lyu, Wei Wang, Jorij Abraham, Ruoyu Wang, Tiffany Bao, Yan Shoshitaishvili, and Adam Doupé. Beyond phish: Toward detecting fraudulent e-commerce websites at scale. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 2566–2583, 2023.

[14] Claudio Carpineto and Giovanni Romano. Learning to detect and measure fake ecommerce websites in search-engine results. WI '17, page 403–410, New York, NY, USA, 2017. Association for Computing Machinery.

[15] Nektarios Leontiadis, Tyler Moore, and Nicolas Christin. Measuring and analyzing search-redirection attacks in the illicit online prescription drug trade. In *Proceedings of the 20th USENIX Conference on Security*, SEC'11, page 19, USA, 2011. USENIX Association.

[16] David Y. Wang, Stefan Savage, and Geoffrey M. Voelker. Cloak and dagger: dynamics of web search cloaking. In *Proceedings of the 18th ACM Conference on Computer and Communications Security*, CCS '11, page 477–490, New York, NY, USA, 2011. Association for Computing Machinery.

[17] Luca Invernizzi, Kurt Thomas, Alexandros Kapravelos, Oxana Comanescu, Jean-Michel Picod, and Elie Bursztein. Cloak of visibility: Detecting when machines browse a different web. In *2016 IEEE Symposium on Security and Privacy (SP)*, pages 743–758, 2016.

[18] Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. Phishfarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1344–1361, 2019.

[19] Penghui Zhang, Adam Oest, Haehyun Cho, Zhibo Sun, RC Johnson, Brad Wardman, Shaown Sarker, Alexandros Kapravelos, Tiffany Bao, Ruoyu Wang, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. Crawlphish: Large-scale analysis of client-side cloaking techniques in phishing. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 1109–1124, 2021.

[20] Xiwen Teoh, Yun Lin, Ruofan Liu, Zhiyong Huang, and Jin Song Dong. PhishDecloaker: Detecting CAPTCHA-cloaked phishing websites via hybrid vision-based interactive models. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 505–522, Philadelphia, PA, August 2024. USENIX Association.

[21] Luca Invernizzi, Stefano Benvenuti, Marco Cova, Paolo Milani Comparetti, Christopher Kruegel, and Giovanni Vigna. Evilseed: A guided approach to finding malicious web pages. In *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, SP '12, page 428–442, USA, 2012. IEEE Computer Society.

[22] Ronghai Yang, Xianbo Wang, Cheng Chi, Dawei Wang, Jiawei He, Siming Pang, and Wing Cheong Lau. Scalable detection of promotional website defacements in black hat SEO campaigns. In *30th USENIX Security Symposium (USENIX Security 21)*, pages 3703–3720. USENIX Association, August 2021.

[23] Google. Custom search json api. Web. Accessed: 2026-01-21. https://developers.google.com/custom-search/v1/overview?hl=ja.

[24] Janome v0.5 documentation(ja). Web. Accessed: 2026-02-18. https://janome.mocobeta.dev/ja/.