

Hidden Below the Landing Page: Tracking Linked Pages and Relation-Centered DOM Summaries for Financial Phishing Detection

Hyeji Heo¹ Dae-Young Park^{1*} Minwoo Kim^{2*}

¹Financial Security Institute (FSI)

²DATUMO INC.

{heohj,mainthread}@fsec.or.kr

mwkim@selectstar.ai

Abstract

Phishing can turn stolen personal information into financial harm, yet some sites hide data collection beyond the landing page. We present a site-level workflow for Korean-language phishing that tracks linked pages and supplies LLMs with relation-centered DOM summaries. The 256 landing URLs treated as phishing in the full collection were labeled primarily by the initial LLM, with human labels taking precedence for reviewed sites. Of these, 161 (62.9%) were detected by at least one VirusTotal engine; 215 of 217 linked URLs (99.1%) had no prior query record, and neither remaining URL was detected. On 70 human-validated landing pages, summarization increased Gemini 2.5 Flash-Lite’s F1 from 0.621 to 0.925, while F1 changes for the other three Gemini models ranged from -0.011 to +0.035. In a six-site case study, data collection was detected only on linked pages, while page-level aggregation retained landing-page impersonation targets more consistently than concatenation. These results highlight the importance of exploration scope and show model-dependent accuracy effects of relation-centered DOM summaries in phishing detection.

1 Introduction

Phishing can turn stolen personal or organizational information into financial gain for offenders and substantial losses for victims. South Korean authorities frame phishing as a digital livelihood crime spanning email phishing, voice phishing, smishing, and QR phishing, in which information theft is used for financial gain (Korea Internet & Security Agency, 2024). One example is institution-impersonation phishing, in which offenders posing as investigators may send a case-lookup link and induce victims to enter personal information as part of a financially motivated fraud (Korean National

Police Agency, National Office of Investigation, 2025).

Such an impersonation site may reproduce the interface and navigation of a prosecution or criminal-justice service. Its landing page may show only the impersonated institution and a case-lookup button, while a linked page collects identity information used later in the fraud process. Similar multi-page flows can also occur on legitimate services, so navigation alone is not evidence of phishing. Whether this structure is copied from the legitimate service or deliberately separates impersonation from collection, landing-page-only analysis can miss the linked page’s data-collection behavior.

Site behavior can span pages, but reputation systems record individual URLs; observing a landing URL does not imply observing its linked paths. Beyond which pages are observed, detection also depends on how their content is represented: LLM-based detectors must interpret DOMs containing duplicated code and official links copied from the impersonated site. We examine (RQ1) how URL-level reputation records differ between landing and linked URLs and (RQ2) how raw versus relation-centered DOM inputs affect performance, cost, and variation across models.

We compare landing- and linked-URL reputation coverage, evaluate DOM representations across four Gemini models on 70 sites, and present a site-level aggregation workflow illustrated by a six-site case study. Our implementation is available on a repository.¹

2 Related Work

Phishing lifecycle studies show that harm concentrates before blocking and that evasion affects blocking time (Oest et al., 2019, 2020a,b). VisualPhishNet and Phishpedia identify impersonated brands visually (Abdelnabi et al., 2020; Lin

*Corresponding authors.

¹https://github.com/hyzzish/FinNLP_2026

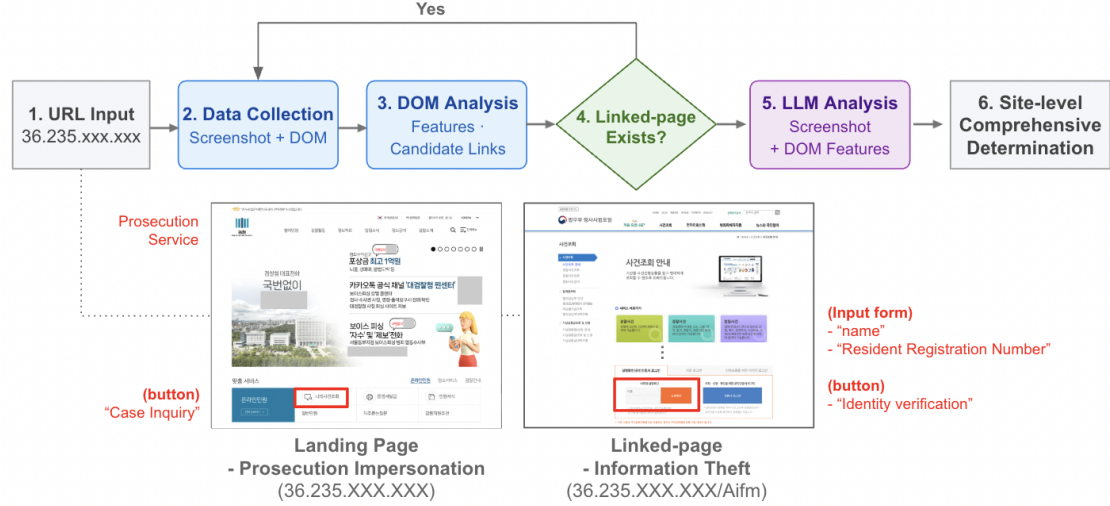


Figure 1: Site-level workflow and example. Linked URLs are extracted, each page is summarized and classified, and page-level analysis results are aggregated.

et al., 2021); PhishIntention additionally clicks likely credential-entry links and analyzes adjacent pages (Liu et al., 2022). We do not claim linked-page exploration as novel, but compare reputation records for landing and linked URLs at a single measurement time.

LLM-based detectors combine URLs, screenshots, HTML, external knowledge, prompting, and reasoning memory (Koide et al., 2024; Li et al., 2024; Cao et al., 2025; Bhargude et al., 2025; Chen et al., 2026). Long-context and web-agent studies further show that representation, encoding, and DOM pruning affect input use (Liu et al., 2024; Gur et al., 2023; Deng et al., 2022; Zhang et al., 2026). Our study compares raw DOMs with summaries that explicitly encode button–input–submission–target relations while holding sites, screenshots, and prompts constant.

3 Site-Level Phishing Analysis Workflow

Figure 1 shows our workflow: extracting linked URLs, summarizing each page’s DOM, and supplying the summaries to an LLM. Hosts may be domains or IP addresses.

We treat a host as one site and save its landing screenshot and rendered DOM. Without clicking, we statically extract destinations from `<a href>` attributes, inline navigation code including addresses indirectly referenced through script constants, and the handlers of interactive elements. Exploration is limited to one hop: we directly access destinations extracted from the landing page, including

external ones, using GET requests, but do not continue exploration from the linked pages. We do not click elements, enter information, submit forms, or perform authentication (Appendix A.1).

The summary has three blocks: **HTML structure** captures inputs, scripts, link targets and navigation logic, and copying or evasion signals; **submission behavior** links buttons to the inputs they read and the destinations they submit to while separating unused code and inputs; and **links** records clickable text, navigation, and destinations (Appendix A.2). A deterministic rule-based extractor generates these summaries without LLM API calls.

For the six-site case study, page-level aggregation uses two LLM calls to analyze the landing and linked pages separately, then supplies only their JSON assessments to a third LLM call for site-level assessment. Input concatenation supplies both pages’ screenshots and DOM summaries together in one LLM call.

4 Data and Experimental Setup

We assembled 693 pages from Korean-language URLScan records observed from July 15–30, 2026 and official sites of impersonated entities. For reachable sites, we captured screenshots and DOMs directly; for offline sites, we used URLScan records. This criterion-based collection is not a random sample and does not estimate the distribution of Korean-language phishing.

We used Gemini 2.5 Flash-Lite for initial labeling under a four-class scheme: phishing,

legitimate, suspicious, or cannot_judge.

For blinded human validation, we sampled 85 sites from pages with valid model predictions, stratified across the four prediction categories. Sampling fractions ranged from 7.5% to 14.9% across strata; the goal was to cover each category, not to estimate population proportions. One reviewer labeled the sampled sites without seeing the model predictions. To form the RQ2 evaluation set, we excluded 14 sites labeled suspicious by the reviewer and one out-of-scope site, leaving 70 sites with determinate human labels: 39 phishing and 31 legitimate.

On these 70 sites, human and initial-model labels agreed on 94.3% under the binary criterion (Cohen’s $\kappa = 0.885$) and 90.0% under the four-class scheme ($\kappa = 0.808$). Among the seven four-class disagreements, the model returned suspicious or cannot_judge in four cases, labeled two reviewer-identified phishing sites as legitimate, and labeled one reviewer-identified legitimate site as phishing. Human labels took precedence for reviewed sites, and the paired evaluation set was fixed before the experimental results were examined.

To construct the RQ1 set from the full collection, we applied the human-label precedence rule and excluded out-of-scope content unrelated to phishing (e.g., gambling, adult, and illegal-trade pages), indeterminate labels (suspicious and cannot_judge), and duplicate hosts. The resulting set contains 256 phishing landing URLs and 217 destinations statically extracted from their buttons and links. Sites impersonating prosecution or criminal-justice services accounted for 193 landing and 184 linked URLs. Most landing labels came from the initial model labeling. In August 2026, we queried only existing VirusTotal results through GET /api/v3/urls/{id} and submitted no new URLs; we treated HTTP 404 as no URL-level query record.

For RQ2, we evaluated four Gemini models on the 70-site human-validated evaluation set described above: 2.5 Flash-Lite, 2.5 Flash, 3.1 Flash-Lite, and 3.5 Flash-Lite. This selection spans model tier and generation, including a non-Lite model and Flash-Lite models from three generations. All models used the same four-class scheme and received the same URLs and screenshots, together with either the full raw DOM or the relation-centered DOM summary. The two 152-character Korean prompts differ only in the terms corresponding to “raw DOM” and “DOM features.” The pri-

mary metric treats only phishing as positive; Appendix A.1 gives full settings.

A separate case study examines six sites built from the same cloned prosecution-service website. Each site exposed one or two linked destinations, but only one linked page per site was reachable and analyzed. Each condition was run once per site; concatenation and aggregation both allowed multiple impersonated targets. All four RQ2 models and the case study used the same summary-extractor version.

The three evaluations cover different parts of the workflow. RQ1 compares URL-level reputation records for landing URLs and extracted linked destinations. RQ2 compares DOM representations on landing pages only. The case study was the only evaluation that included linked-page collection and site-level analysis, including page-level aggregation, and was limited to six sites.

5 Results

5.1 URL Reputation Coverage

Of the 256 landing URLs, 25 (9.8%) had no prior query record. Of the remaining 231, 161 were detected by at least one engine (69.7% among queried URLs and 62.9% overall). By contrast, 215 of 217 linked URLs (99.1%) had no record, and neither remaining URL was detected. One was a genuine government URL copied into a phishing page, so relationally extracted linked URLs are not necessarily malicious. These results compare URL-level observation coverage, not classifier performance or host-level protection.

Among queried landing URLs, the 189 impersonating prosecution or criminal-justice services had a mean of 1.01 detecting engines (median 1, maximum 4), versus 6.14 (median 2, maximum 23) for the other 42 phishing URLs. Detection of this group was therefore concentrated among a small number of engines. Appendix A.3 provides the full reputation distributions.

5.2 DOM Representation and Model Comparison

RQ2 provides a paired comparison of raw DOMs and summaries on the same 70 human-validated landing pages. The reported F1 scores describe the evaluated models’ performance on these sites, but these scores alone would not establish how well a phishing detection service using these models would perform on sites encountered in practice.

Summarization substantially improved Gemini 2.5 Flash-Lite’s F1 from 0.621 to 0.925, compensating for its poor performance with raw DOM inputs. False negatives decreased from 21 to 2, while false positives increased from 1 to 4. The other three Gemini models already achieved high F1 scores with raw DOM inputs (0.918–0.975) and showed smaller changes under summarization, between -0.011 and +0.035. Table 1 reports the results for all four models.

Model	Raw	Summary	Δ
2.5 Flash-Lite	0.621	0.925	+0.304
2.5 Flash	0.918	0.907	-0.011
3.1 Flash-Lite	0.975	0.975	0.000
3.5 Flash-Lite	0.927	0.962	+0.035

Table 1: Strict F1 by DOM representation.

For Gemini 2.5 Flash-Lite, the exact McNemar comparison had 19 raw-only versus 3 summary-only errors ($p = 8.55 \times 10^{-4}$; Holm-adjusted across the four per-model comparisons of the two input representations, $p = 0.00342$). Bootstrap 95% F1 intervals were [0.464, 0.761] for raw DOMs and [0.861, 0.977] for summaries.

Across the four Gemini models evaluated, the performance gap, measured as the difference between the highest and lowest F1 scores, narrowed from 0.354 to 0.068. This narrowing largely reflected the substantial improvement in Gemini 2.5 Flash-Lite rather than consistent gains across all four models.

A copied Naver page illustrates this model-dependent pattern. With raw DOM inputs, 2.5 Flash-Lite cited canonical=naver.com and official CDN and service links as evidence of legitimacy. With the summary, its explanation instead emphasized the mismatch between the accessed host and the official domain as evidence of impersonation. By contrast, 2.5 Flash and 3.1 Flash-Lite already interpreted the copied official resources as evidence of phishing construction in the raw-DOM condition.

Although these results do not establish a general accuracy benefit from summarization, they demonstrate that summarization can substantially improve phishing detection performance for some models. Because all four evaluated models came from a single vendor, further validation is needed to determine whether these findings extend to models from other vendors.

These model-dependent effects motivate evaluat-

Model	F1	Cost	Latency
2.5 F-Lite	0.925	\$0.028	4.4 s
2.5 Flash	0.907	\$0.136	16.2 s
3.1 F-Lite	0.975	\$0.072	3.3 s
3.5 F-Lite	0.962	\$0.103	2.7 s

Table 2: Summary-condition F1, total model-inference cost for 70 sites, and mean latency.

ing the model and input representation jointly, with attention to accuracy, cost, and latency.

In terms of efficiency, DOM summarization reduced mean input length by a factor of approximately 23, from 42,314 to 1,835 tokens. For 2.5 Flash-Lite, the 70-site model-inference cost fell from \$0.303 to \$0.028, although mean latency rose from 3.9 to 4.4 seconds, alongside an increase in output length from 245 to 554 tokens. Across all four models, summaries reduced model-inference cost, but mean latency increased. Table 2 compares summary-condition performance and operating cost; detailed performance and error results are reported in Appendix A.4.

5.3 Case Study: Landing and Linked Pages

In the example shown in Figure 1, the landing page mimicked a prosecution service and linked to a criminal-justice portal imitation requesting a name and identification number. The summary found navigation but no collection action on the landing page; on the linked page, it identified the verification button’s input fields and submission target (Appendix A.2). Table 3 compares four conditions using Gemini 2.5 Flash-Lite on the six sites. One reviewer manually assessed target retention, treating abbreviations and alternative names for the same entity as equivalent.

Condition	Phish.	Collect.	Landing
Landing only	6/6	0/6	–
Linked only	6/6	6/6	–
Concatenated	6/6	6/6	2/6
Page-level aggregate	6/6	6/6	6/6

Table 3: Page-scope comparison. “Collect.” is collection-behavior detection; “Landing” is retention of the landing-page target.

All four conditions classified all six sites as phishing; this case study therefore showed no classification improvement. Linked-page analysis revealed data-collection behavior absent from the landing pages; aggregation and concatenation both detected it in all six cases. Aggregation retained

the landing-page impersonation target in six cases versus two for concatenation. Under the stricter criterion of retaining both pages' targets, the counts were six and one, respectively. The aggregate outputs combined targets already identified in the page-level JSON assessments. Thus, linked-page analysis exposed additional behavioral evidence, while aggregation more consistently retained previously extracted impersonation targets in these cases.

6 Discussion and Conclusion

Landing URLs commonly had reputation records, while linked URLs usually did not; this reflects URL-level observation scope, not failure by a specific service. Relation-centered summaries reduced model-inference cost and compensated for one model's raw-DOM degradation. In the six-site case study, page-level aggregation retained previously extracted impersonation targets more consistently than concatenation.

Further study should test the generality and sources of the representation effects. Comparisons with other vendors' models should use the same evaluation sites, URLs, screenshots, and prompts, with untruncated raw DOMs, to assess performance and cost differences across models and vendors. Length-matched pruned DOMs, basic-cleaning baselines, and summaries without explicit relational encoding would help distinguish compression, noise removal, and relational contributions. Ablations of the HTML structure, submission behavior, and link blocks would further assess individual components.

Broader validation should expand human review of RQ1 labels and repeat RQ2 on a larger, independently sampled set selected without using the initial model's predictions. Multiple reviewers should label sites independently under a shared protocol, with agreement reported and disagreements adjudicated; deployment-performance estimates would require sampling representative of the intended setting. Aggregation and concatenation should also be compared on a larger set of multi-page sites from diverse phishing campaigns and legitimate services with similar navigation and forms. Measuring both cross-page evidence retention and site-level classification, including false-positive rates, would test whether evidence preservation translates into better detection.

Exploration could also extend to selected dynamic and multistep flows beyond static one-hop

access. Such extensions should weigh additional evidence and site-level detection gains against request volume and risks of interacting with live phishing infrastructure. Explicit stopping rules and restrictions on data entry and form submission would be needed to define a safe exploration boundary.

Together, these findings motivate analyzing directly linked pages and considering relation-centered DOM summaries as an input option for LLM-based phishing detection, with model-dependent accuracy effects.

Limitations

Most RQ1 labels came from the initial model labeling rather than independent human review. The collection is dominated by sites impersonating prosecution or criminal-justice services (193/256 landing and 184/217 linked URLs); 186 of this group's 193 distinct hosts (96.4%) were submitted by one URLScan API submitter. Thus, the characteristics of this group cannot be separated from submitter-specific visibility, and sample share should not be interpreted as prevalence. Moreover, 183 linked URLs share their landing hosts, and URL-level records do not reveal host-level protection.

The RQ2 evaluation set contains 70 human-validated sites sampled across the initial model's prediction categories, rather than to represent an intended deployment setting. Ground truth reflects one blinded reviewer, and all evaluated models came from a single vendor. These constraints limit the generalizability of the observed representation effects.

The raw-DOM versus full-summary comparison evaluates the summary representation as a whole. It does not isolate the contributions of input compression, noise removal, or explicit button-input-submission-target relations. The observed performance gains therefore cannot be attributed to relational encoding alone.

The six-site case study is illustrative, covers sites built from the same cloned prosecution-service website, and includes one run per condition. Rule-based summaries may omit unmodeled evidence. Exploration covers only statically extracted landing-page destinations directly accessible within one hop; dynamic interaction, authentication, and multistep input are excluded. Obfuscated navigation and redirect-dependent flows may also be missed.

Ethical Considerations

In the experiments reported in this paper, we entered no values, submitted no forms or VirusTotal URLs, and made no POST requests or probes. Six case-study sites were accessed only by sequential GET requests. We stored public screenshots and code, accessed no victim data, and redacted phone numbers and QR codes in Figure 1.

References

- Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. 2020. VisualPhishNet: Zero-day phishing website detection by visual similarity. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*.
- Atharva Bhargude, Ishan Gonehal, Dave Yoon, Sean O'Brien, Kaustubh Vinnakota, Chandler Haney, Aaron Sandoval, and Kevin Zhu. 2025. Adaptive linguistic prompting (ALP) enhances phishing webpage detection in multimodal large language models. In *Proceedings of the Fourth Workshop on NLP for Positive Impact*, pages 77–85.
- Tri Cao, Chengyu Huang, Yuexin Li, Huilin Wang, Amy He, Nay Oo, and Bryan Hooi. 2025. PhishAgent: A robust multimodal agent for phishing webpage detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*.
- Xuan Chen, Hao Liu, Tao Yuan, Mehran Kafai, Piotr Habas, and Xiangyu Zhang. 2026. MemoPhishAgent: Memory-augmented multi-modal LLM agent for phishing URL detection. In *Proceedings of the ACL Industry Track*, pages 1182–1196.
- Xiang Deng, Prashant Shiralkar, Colin Lockard, Binxuan Huang, and Huan Sun. 2022. DOM-LM: Learning generalizable representations for HTML documents. *arXiv preprint arXiv:2201.10608*.
- Google. 2026. [Gemini developer api pricing](#). Accessed August 9, 2026.
- Izzeddin Gur, Ofir Nachum, Yingjie Miao, Mustafa Safdari, Austin Huang, Aakanksha Chowdhery, Sharan Narang, Noah Fiedel, and Aleksandra Faust. 2023. Understanding html with large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 2803–2821.
- Takashi Koide, Hiroki Nakano, and Daiki Chiba. 2024. ChatPhishDetector: Detecting phishing sites using large language models. *IEEE Access*, 12:154381–154400.
- Korea Internet & Security Agency. 2024. Domestic and international phishing response status and implications. Technical Report KISA Insight 2024 Vol. 07, Korea Internet & Security Agency. KISA Insight 2024 Vol. 07, report in Korean; title translated from Korean.
- Korean National Police Agency, National Office of Investigation. 2025. Voice phishing: Would you become the protagonist of a meticulous script? Press release, April 28, 2025, in Korean; title translated from Korean.
- Yuexin Li, Chengyu Huang, Shumin Deng, Mei Lin Lock, Tri Cao, Nay Oo, Hoon Wei Lim, and Bryan Hooi. 2024. KnowPhish: Large language models meet multimodal knowledge graphs for enhancing reference-based phishing detection. In *33rd USENIX Security Symposium*, pages 793–810.
- Yun Lin, Ruofan Liu, Dinil Mon Divakaran, Jun Yang Ng, Qing Zhou Chan, Yiwen Lu, Yuxuan Si, Fan Zhang, and Jin Song Dong. 2021. Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages. In *30th USENIX Security Symposium*.
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.
- Ruofan Liu, Yun Lin, Xianglin Yang, Siang Hwee Ng, Dinil Mon Divakaran, and Jin Song Dong. 2022. Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach. In *31st USENIX Security Symposium*.
- Adam Oest, Yeganeh Safaei, Adam Doupé, Gail-Joon Ahn, Brad Wardman, and Kevin Tyers. 2019. PhishFarm: A scalable framework for measuring the effectiveness of evasion techniques against browser phishing blacklists. In *2019 IEEE Symposium on Security and Privacy*.
- Adam Oest, Yeganeh Safaei, Penghui Zhang, Brad Wardman, Kevin Tyers, Yan Shoshitaishvili, Adam Doupé, and Gail-Joon Ahn. 2020a. PhishTime: Continuous longitudinal measurement of the effectiveness of anti-phishing blacklists. In *29th USENIX Security Symposium*.
- Adam Oest, Penghui Zhang, Brad Wardman, Eric Nunes, Jakub Burgis, Ali Zand, Kurt Thomas, Adam Doupé, and Gail-Joon Ahn. 2020b. Sunrise to sunset: Analyzing the end-to-end life cycle and effectiveness of phishing attacks at scale. In *29th USENIX Security Symposium*.
- Jiayuan Zhang, Kaiquan Chen, Zhihao Lu, Enshen Zhou, Qian Yu, and Jing Zhang. 2026. Prune4Web: DOM tree pruning programming for web agent. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

A Experimental Details and Additional Results

A.1 Collection, Model, and Input Configuration

Linked-page exploration was limited to one hop from the landing page. Greater depth can increase requests with the branching factor, thereby increasing contact with live phishing infrastructure; moreover, paths gated by authentication or multistep input cannot be reached through direct navigation alone.

We directly accessed the statically extracted destinations using GET requests without clicks or form submissions and captured pages sequentially. For each capture, we waited up to six seconds for network idle and an additional 1.5 seconds for rendering. Paths beyond the linked pages, dynamic events and routing, obfuscated navigation, authentication, and multistep interaction were excluded.

In RQ1, the 217 extracted destinations were used for reputation lookup; full linked-page collection and site-level analysis were limited to the six case-study sites. In all six, the reachable page collecting personal information was one hop from the landing page.

Item	Setting
2.5 F Lite	gemini-2.5-flash-lite / v001
2.5 Flash	gemini-2.5-flash / v001
3.1 F Lite	gemini-3.1-flash-lite /
	3.1-flash-lite-05-2026
3.5 F Lite	gemini-3.5-flash-lite /
	3.5-flash-lite-07-2026
Temperature / top-p	0 / API default
Screenshot	Playwright 1280 × 800, full_page=True, maximum height 5,000 px
Raw DOM	Provided in full without truncation
Summary generation	Deterministic rule-based extraction; no LLM API calls
Output	Four-way decision, one run per condition

Table 4: Model and input settings.

The two Korean prompts were 152 characters long and differed only in the phrases corresponding to “raw DOM” and “DOM features.” The experiments used the Korean originals; their shared template is translated below.

You are a phishing-site analyst. Examine the given screenshot, identify the institution and its website address (URL or IP), check whether it matches the given address, and analyze the given {DOM input} to determine whether the site is a phishing site. Include the evidence supporting your phishing decision.

A.2 Relation-Centered DOM Summary

The counts show how many field types each block contains; the summary is not a fixed-length feature vector. Field groups repeat when a page contains

Block	Unit	Types	Main content
HTML structure	Page	28	Inputs, scripts, identity, link targets, navigation logic, copying and evasion
Submission behavior	Action	13	Trigger, function, target, status, fields, unused inputs
Links	Element	8	Text, navigation, destination, external status, empty targets

Table 5: Components of the relation-centered DOM summary.

multiple submission actions, and an empty list represents the absence of an action. The following excerpts are English translations of values extracted from the Korean case-study site.

Landing-page linked URL

```
{
  "label": "Case Lookup",
  "via": "href",
  "dst": "/Ahyn",
  "ext": false
}
```

Linked-page submission behavior

```
{
  "trigger": "Identity Verification",
  "kind": "js",
  "handler": "f_guestLogin",
  "submission": "/Ahyinpost24",
  "submission_same_host": true,
  "life": "alive",
  "fields": [
    {"title": "Name", "type": "text"},
    {"title": "ID front", "type": "text"},
    {"title": "ID back",
     "type": "password"}
  ]
}
```

A.3 Reputation Coverage

Layer	No record	0	1	2–5	6+	Total
Landing	25	70	104	43	14	256
Linked	215	2	0	0	0	217

Table 6: Distribution of the number of detecting Virus-Total engines.

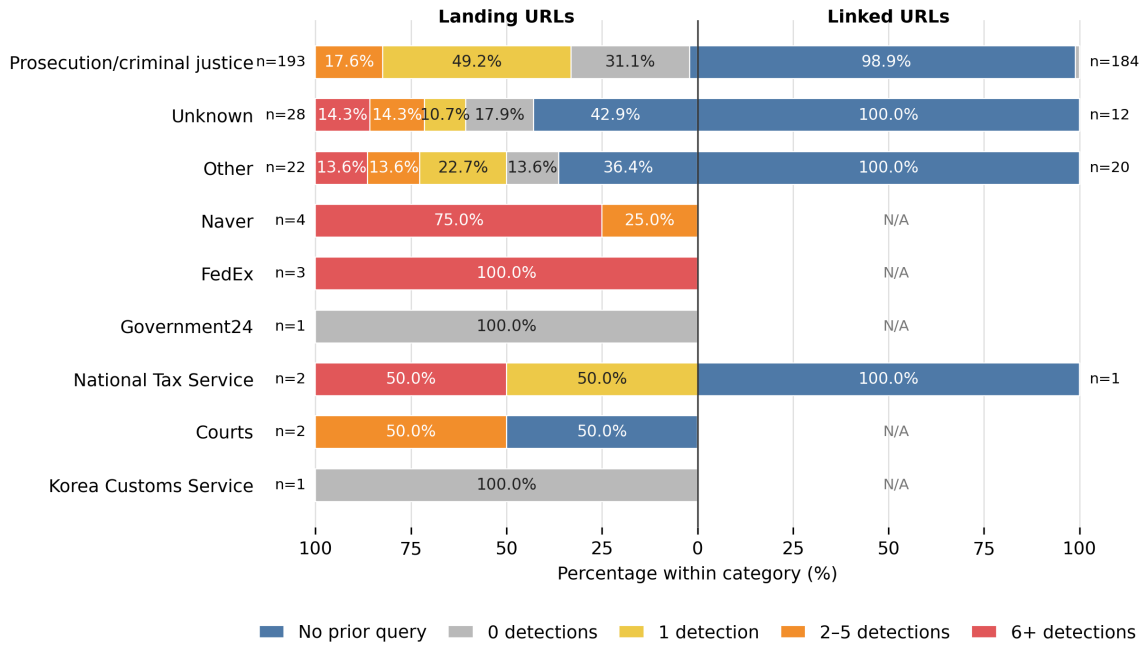


Figure 2: VirusTotal observation and detection distributions for landing and linked URLs by impersonated entity. Each bar is normalized within its category; internal labels show percentages, and n is the number of URLs. Counts of two to five detecting engines are grouped into one interval. Categories without extracted linked URLs are marked N/A.

Landing category	Queried	Mean	Median	Max.
Prosecution/criminal justice	189	1.01	1	4
Other phishing	42	6.14	2	23

Table 7: Detecting-engine counts among queried landing URLs.

A.4 Performance, Cost, Latency, and Tests

Model	Strict raw	Strict summary	Expanded raw	Expanded summary
2.5 F Lite	0.621	0.925	0.800	0.886
2.5 Flash	0.918	0.907	0.867	0.876
3.1 F Lite	0.975	0.975	0.897	0.929
3.5 F Lite	0.927	0.962	0.897	0.916

Table 8: F1 under the strict (only phishing is positive) and expanded-positive (phishing and suspicious are positive) criteria. Across-model strict-F1 ranges are 0.354 (raw) and 0.068 (summary). Main-text confidence intervals use 2,000 paired percentile-bootstrap resamples (seed=42).

Costs were calculated using public Gemini API prices accessed on August 9, 2026 (Google, 2026). Rule-based summary generation incurred no LLM API cost.

For 2.5 Flash-Lite, summary-condition errors comprised two false negatives for individual impersonation without official comparison sites and four false positives for legitimate free-hosting, IP-address, or landing sites.

Input	2.5 F Lite	2.5 Flash	3.1 F Lite	3.5 F Lite
Raw DOM	21 / 1	0 / 7	0 / 2	1 / 5
Summary	2 / 4	0 / 8	0 / 2	1 / 2

Table 9: False negatives / false positives under the strict criterion.

Model	Raw DOM	Summary
2.5 F Lite	245 / \$0.303 / 3.9 s	554 / \$0.028 / 4.4 s
2.5 Flash	490 / \$0.974 / 14.7 s	559 / \$0.136 / 16.2 s
3.1 F Lite	198 / \$0.761 / 2.5 s	379 / \$0.072 / 3.3 s
3.5 F Lite	169 / \$0.918 / 2.3 s	366 / \$0.103 / 2.7 s

Table 10: Output tokens / total model-inference cost for 70 sites / mean latency. Mean inputs were 42,314 tokens (raw) and 1,835 (summary), including 258 screenshot tokens per page.