INTERNATIONAL RESEARCH JOURNAL OF ENGINEERING & APPLIED SCIENCES
ISSN: 2322-0821(O)
ISSN: 2394-9910(P)
VOLUME 13 ISSUE 4
Oct 2025 - Dec 2025
www.irjeas.org

Research Article

ADEPT: An Autonomous, LLM-Powered Agent for Dynamic, Reference-Based Phishing Detection

***Ashwarya Singh*[1*], *Kalpana Mishra*[2]**

[1] Research Scholar, Dept. of Computer Science Engineering, JNCT, Bhopal, India
jpahwary1999@gmail.com
[2] Asst. Professor, Dept. of Computer Science Engineering, JNCT, Bhopal, India
kalpana.cse@jnctbhopal.ac.in

*Corresponding Author jpahwary1999@gmail.com

DOI–10.55083/irjeas.2025.v13i04009

Abstract- Conventional phishing detection systems, which predominantly rely on static blacklists and rigid, signature-based heuristics, are increasingly ineffective against the dynamic and polymorphic nature of modern phishing attacks. Reference-based detection, which verifies a webpage's authenticity against a known brand's identity, offers a more robust approach but is critically hampered by the need to maintain a comprehensive and constantly updated knowledge base. This paper addresses this fundamental challenge by introducing ADEPT (Autonomous Dynamic Agent for Phishing Threat), a novel framework architected around a sophisticated Large Language Model (LLM) acting as an autonomous agent. We first present a granular error analysis of a state-of-the-art dynamic reference-based system, DynaPhish, revealing its brittleness and critical failures stemming from rigid logic and dependency on external APIs. To overcome these deficiencies, we designed and implemented the ADEPT framework, which equips an LLM agent with a multi-modal perception pipeline and a toolkit for real-time information retrieval from the web. The agent mimics human cognitive processes to dynamically investigate suspicious webpages, analyze visual and textual content, and reason about a brand's identity without

relying on a pre-existing static knowledge base. Through a series of rigorous experiments on a balanced dataset of 400 phishing and benign samples, the ADEPT framework, utilizing the GPT-4 model, achieved a phishing detection accuracy of 0.945. This represents a significant improvement over both the DynaPhish baseline (0.499 accuracy) and simpler LLM-based methods, empirically validating that an autonomous, agent-based approach provides a more resilient and effective solution to the pervasive threat of phishing.

Keywords—**Phishing Detection, Cybersecurity, Autonomous Agents, Large Language Models (LLM), Generative AI, Reference-Based Detection, Dynamic Web Analysis.**

I. INTRODUCTION

The global digital ecosystem's rapid expansion has created unprecedented opportunities for communication, commerce, and innovation [3]. However, this hyper-connectivity has also cultivated a fertile ground for malicious actors, with phishing emerging as one of the most persistent and damaging threats [1]. Defined as a social engineering attack that aims to illicitly acquire sensitive information by impersonating a trustworthy entity, phishing serves as the primary vector for an estimated 90% of all data breaches worldwide [2]. The problem is particularly acute in rapidly digitalizing economies, where a large influx of new internet users creates a vast pool of potential targets [4], [5]. The financial sector remains a primary target, with threat actors meticulously crafting counterfeit websites of major financial institutions to harvest user credentials [6], leading to substantial economic losses [7].

Traditional defenses against phishing have proven increasingly inadequate. Conventional detection systems predominantly rely on static, signature-based methodologies such as blacklisting known malicious URLs and keyword-based content filtering [8]. These techniques are fundamentally reactive and suffer from a critical inability to detect novel or "zero-day" phishing attacks. Cybercriminals can easily circumvent these static defenses by rapidly registering new domains, using URL obfuscation, and constantly altering their attack templates, creating a significant temporal gap between the launch of a campaign and its detection.

This research addresses the inadequacy of these static methods by proposing a paradigm shift towards a dynamic, intelligent, and autonomous detection framework. The primary aim of this work is to design, implement, and rigorously evaluate a novel system for reference-based phishing detection that leverages the advanced reasoning and information-processing capabilities of a sophisticated Large Language Model (LLM) agent. We introduce the ADEPT (Autonomous Dynamic Agent for Phishing Threat) framework, a system designed to overcome the critical limitations of existing solutions by eschewing reliance on a static knowledge base. Instead, ADEPT empowers an autonomous agent to perform real-time, dynamic web analysis, mimicking human cognitive processes of information retrieval, contextual analysis, and decision-making.

This paper is structured as follows. Section II provides background on phishing detection methodologies and presents a deep empirical analysis of the failures of an existing dynamic system, establishing the motivation for our work. Section III details the complete design and architecture of the proposed ADEPT framework. Section IV describes the specific implementation choices and strategies for the framework's core components. Section V presents the comprehensive experimental evaluation, including brand recognition and end-to-end phishing classification performance. Section VI discusses the results, provides a comparative analysis against baseline models, and Section VII acknowledges the study's limitations and outlines future work. Finally, Section VIII concludes the paper, summarizing its key contributions to the field of cybersecurity.

II. BACKGROUND AND PROBLEM ANALYSIS

The evolution of phishing detection has been a continuous effort to outpace the adaptive strategies of attackers. Methodologies have progressed from simple static lists to complex machine learning models. However, even advanced approaches exhibit critical vulnerabilities when faced with the dynamic nature of the web. This section briefly reviews these methodologies and provides a deep, empirical investigation into the failures of a modern reference-based system, which serves as the primary motivation for our agent-based approach.

A. An Overview of Phishing Detection Methodologies

Phishing detection techniques are broadly categorized into four groups: Blacklist-based, Heuristic-based, Visual Similarity, and Machine Learning approaches [12]. Blacklist-based methods, employed by services like Google Safe Browsing and PhishTank, are precise but fundamentally reactive, failing against novel threats [15]. Heuristic-based methods analyze URL and content features for suspicious patterns but often suffer from high false-positive rates. Machine learning introduced a significant leap in adaptability, with models trained to extract predictive features from URLs using NLP techniques [16] or to analyze visual content. The latter

gave rise to reference-based detection, where a webpage's visual elements, particularly brand logos, are compared against a trusted reference. Systems like Phishpedia [13] and PhishIntention [14] use deep learning for logo identification and intent inference, demonstrating high accuracy but relying on a comprehensive, pre-compiled knowledge base of protected brands.

B. Empirical Investigation of a Dynamic Reference-Based System

The critical bottleneck for reference-based systems is the maintenance of their knowledge base. The DynaPhish system was developed to address this by dynamically expanding its knowledge base using web search and external APIs [9]. Due to its modern design, we selected DynaPhish for a detailed empirical examination to understand the practical challenges of this paradigm.

To isolate its core reference-based capabilities, we reproduced a limited implementation of DynaPhish, disabling its web interaction components (CRP classification) and starting with a nearly empty knowledge base containing only two brands. We then processed the Phishing 3k dataset, containing

approximately 3,310 samples across 340 brands. The system achieved a recall rate of only 40%, though it successfully expanded its knowledge base to 216 brands. This low recall prompted a granular error analysis to identify the root causes of its detection failures.

C. Granular Error Analysis of DynaPhish

Our deep dive revealed that DynaPhish's failures were not random but stemmed from systemic flaws in its rigid, programmatic logic.

1) *Logo Detection and Recognition Failures*: A primary point of failure was the initial logo detection. In our test, 87 out of 3,310 samples were incorrectly classified as benign because the system either failed to detect a logo or cropped the wrong image element. This could be due to unconventional logo placement or the complete absence of a logo on the phishing page. An incorrect crop invariably leads to a failure in all subsequent recognition and validation steps, causing an immediate false negative.
2) *Systemic Representation Failure*: The most critical flaw lies in DynaPhish's multi-step Representation Validation process for knowledge expansion. The process is as follows: a) The logo is cropped from a screenshot. b) Google's logo detection service identifies the brand. c) The brand name is used to query Google Search. d) The top five search results are filtered. e) A web driver visits the filtered links, retrieves their logos, and compares them to the original cropped logo. f) If the similarity score exceeds a threshold, the brand is added to the knowledge base.

This entire chain is highly brittle. A key issue is the over-reliance on the Google Logo Detector; any inaccuracy at this stage invalidates the entire process. Furthermore, the mechanism for retrieving reference logos is fundamentally flawed, leading to several distinct failure modes:

a) **Incomplete Brand Representation:** The system retrieves only the current, official logos from a brand's primary website. It cannot account for logo variants,

such as older versions or sub-brand logos. This was starkly evident in the detection of AT&T phishing samples. Of 103 samples using various legitimate AT&T logo variants, DynaPhish correctly classified only two because it only had the single logo from att.com as a reference, causing all other variants to fail the validation threshold.

b) **Overly Restrictive Filtering:** The filters applied to the Google search results, intended to reduce noise, can inadvertently block access to legitimate reference logos. For example, during the analysis of Instagram phishing samples, DynaPhish failed to detect 113 out of 119 samples. Although the logo was correctly identified, the system's filters automatically excluded search results containing the domain 'instagram.com', preventing the retrieval of the correct reference logo and causing a complete breakdown in both detection and knowledge expansion.

c) **Technical Edge Cases:** The system's automated web driver can be easily thwarted by modern web security measures. During the validation process for the brand "Bitkub," the driver was

consistently blocked by Cloudflare security verification pages on the legitimate websites. This prevented the successful retrieval of any reference logos, leading to a 100% failure rate for all Bitkub samples.

These empirical findings demonstrate that while the goal of dynamic knowledge expansion is correct, the rigid, step-by-step algorithmic approach of systems like DynaPhishis too fragile for the complexities of the real web. It lacks the reasoning and adaptability to handle ambiguity, variation, and unexpected obstacles. This motivates the need for a new architecture centered on an intelligent agent that can reason and plan dynamically.

III. THE ADEPT FRAMEWORK: DESIGN AND ARCHITECTURE

To address the identified limitations of existing systems, we designed the ADEPT framework, an integrated system architected around an autonomous LLM agent. The design philosophy is to mimic the human cognitive approach to investigating a suspicious webpage—gathering multi-modal information, cross-referencing it with external knowledge, and making a reasoned judgment. The framework consists of three core components: the Information Preprocessing module, the Generative Agent core, and the Domain Checker module.

A. Rationale for a New LLM-Based Approach

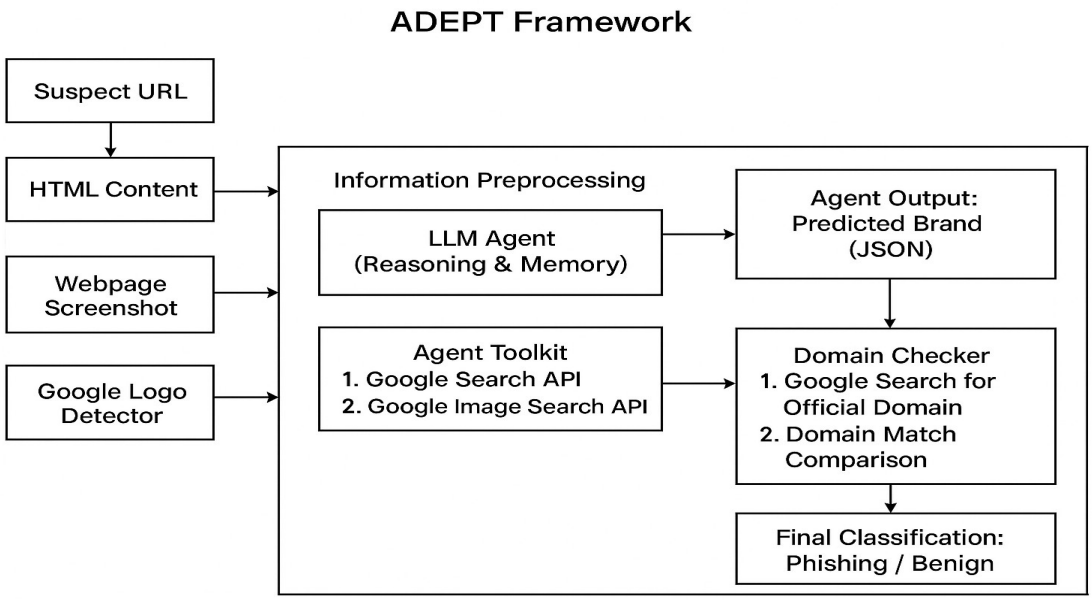
The failures of DynaPhish highlight that the core challenge in reference-based detection is not just data retrieval, but intelligent information processing and reasoning. LLMs, with their advanced capabilities in understanding context, handling ambiguity, and planning sequences of actions, are uniquely suited to this task [17], [25]. By framing the detection problem as an investigative task for an autonomous agent, we can leverage these strengths. The agent can dynamically decide what information it needs, how to acquire it, and how to synthesize it to

reach a conclusion, replacing the brittle, hard-coded logic of previous systems with flexible, goal-oriented reasoning.

B. ADEPT System Architecture

The overall architecture of ADEPT is depicted in Fig. 1. The system operates as a pipeline that takes a suspect URL as input and outputs a binary classification (Phishing/Benign). The process begins with preprocessing the webpage's content, which is then fed to the agent. The agent utilizes its toolkit to gather more information before making a final brand prediction, which is then passed to the Domain Checker for final verification.

Fig. 1. Overview of the Proposed ADEPT Framework Architecture.



C. Information Preprocessing Module

To provide the agent with a comprehensive and digestible understanding of the webpage, this module processes the raw web content that a user interacts with: the URL, text, and visual elements. Given a URL, the system first extracts the full HTML and a screenshot.

1) **HTML Processing:** The raw HTML is parsed and refined to fit within the token limits of the LLM. Only crucial elements for context are retained, including the page title, text rendered on the frontend, and the structure of input forms and buttons.
2) **Visual Content Analysis:** A logo is first extracted from the screenshot using a cropping algorithm. To translate this visual information into textual understanding for the agent, we use two tools:

a) **Google Logo Detector:** This provides a quick, initial prediction of the brand associated with the cropped logo.

b) **GPT-4 Vision (GPT-4V):** Both the cropped logo and the entire screenshot are passed to the GPT-4V model [17]. This leverages its advanced Optical Character Recognition (OCR) and reasoning capabilities to provide a rich, descriptive analysis of the visual content, including text within images and the overall page layout, identifying the target brand from a holistic visual perspective.

This preprocessing step effectively translates the multi-modal, explicit information on the page into a structured, text-based format imbued with implicit meaning, which serves as the initial context for the agent.

D. The Generative Agent Core

The heart of the framework is an agent simulated using the GPT-4 model, leveraging its native support for function calling and conversational reasoning. The agent's

operation is guided by a carefully crafted system prompt that defines its role, objectives, and constraints.

1) **Prompt Engineering:** The agent is instructed to act as an expert cybersecurity assistant with strong reasoning skills. It is tasked to be truthful and to ground all its claims and decisions in the information provided or gathered through its tools.
2) **Objective:** The agent's final goal is to determine the single brand most likely associated with the webpage. If it cannot confidently identify a known brand, it must report that and provide clear reasons for its decision.
3) **Constraints:** To manage runtime costs and prevent infinite loops, the agent is limited to a maximum of five tool calls before it must provide its final output in a structured JSON format.

E. Agent Toolkit and Function Calls

The agent is equipped with a toolkit of functions it can call to dynamically expand its knowledge. The LLM does not execute the functions itself but generates a JSON object containing the function name and arguments to call [17]. Our framework provides two primary tools:

1. **Google Search:** The agent can construct and execute search queries based on the information it has, such as the page title or the brand name suggested by the vision models. The search results (titles, snippets, and URLs) are returned to the agent and appended to its conversational history, forming an enriched context for its next reasoning step.
2. **Google Image Search:** This function allows the agent to verify a brand's logo. It takes a search query and returns a list of relevant images. These images are then downloaded, and a similarity check is performed against the original cropped logo using a pretrained

   computer vision model. The aggregated similarity scores are added to the agent's history, providing strong evidence for logo validation.

Through iterative use of these tools, the agent dynamically builds a comprehensive understanding of the webpage's claimed identity and its relation to legitimate brands on the internet.

F. Domain Checker Module

Once the agent concludes its investigation and outputs its final brand prediction, the result is passed to the Domain Checker. This module performs the final verification step. It takes the brand name identified by the agent as a query for Google Search and compiles a reference list of official domains from the top search results. If the domain of the original suspect webpage matches any domain in this dynamically generated list, the page is classified as benign; otherwise, it is classified as phishing. This strategy completely circumvents the need for a static knowledge base and the complex, failure-prone Representation Validation process of DynaPhish.

IV. IMPLEMENTATION DETAILS

The practical implementation of the ADEPT framework was guided by the strategic

separation of processes for static information translation and dynamic knowledge expansion. This design choice reflects the distinction between the explicit information directly present on a webpage and the implicit knowledge required to understand its context and authenticity.

A. A Pipeline for Knowledge Transformation

Our implementation is designed as a pipeline that transforms raw, explicit webpage content (text, images, URL) into actionable, implicit knowledge for the agent. This is achieved by categorizing the available tools based on their role in this process.

1) **Tools for Static Information Translation (Preprocessing):** The Google Logo Detector and the GPT-4 Vision (GPT-4V) model are used exclusively during the initial preprocessing stage. Their purpose is to analyze the static content of the webpage and translate it into a structured, text-based format. For instance, an image of a bank's logo (explicit information) is translated by GPT-4V into the text "This is the logo for Brand X" (implicit knowledge). This provides the agent with a rich, foundational understanding before it begins its own dynamic investigation.
2) **Tools for Dynamic Knowledge Expansion (Agent Toolkit):** Google Search and Google Image Search are provided as part of the agent's active toolkit. These are not used on the static content of the page itself. Instead, they facilitate the agent's real-time exploration of the wider internet to gather external, corroborating evidence. This allows the agent to actively seek new information to validate or refute hypotheses about the webpage's authenticity, enabling a truly dynamic investigation.

B. Implementation of the Domain Verification Module

The Domain Checker is the final arbiter in the classification process. Its implementation was optimized to reliably identify the legitimate domain(s) of a target brand while being resilient to common variations in corporate web presences.

Our strategy employs a carefully designed domain-matching mechanism that checks for both top-level (e.g., brand.com) and second-level (e.g., subdomain.brand.com) domain matches. To generate the reference list of legitimate domains, we use the raw brand name identified by the agent, enclosed in quotation marks (e.g., "State Bank of India"), as an exact-match query to the Google Search API. This precise query method ensures that the search engine returns the most authoritative webpages associated with the target brand in its top results. From these results, we extract the display links to compile the domain matching list.

After experimentation, we determined that extending the length of this reference list was crucial for reducing false positives caused by domain variants (e.g., country-specific sites like brand.co.uk or sub-brands like support.brand.com). As shown in our experiments in the next section, increasing the list from a single domain to 10 significantly improved performance. We also considered implementing a check for URL redirections.

However, this proved to be technically challenging and computationally expensive, as many sites use delayed or JavaScript-based redirections that require a full browser engine like Selenium to detect reliably. Given the significant increase in runtime cost, we opted for a more efficient solution. Our final implementation uses a reference list of **10 domains** compiled from the top search results, without a redirection check, representing a balanced trade-off between accuracy and operational efficiency.

V. EXPERIMENTAL EVALUATION

To quantitatively assess the efficacy of the ADEPT framework, we conducted a multi-faceted empirical evaluation structured into two sequential segments: a foundational test of the agent's Brand Recognition capabilities and a holistic, end-to-end evaluation of its Phishing Classification performance.

A. Datasets and Experimental Setup

For our experiments, we used two well-established and diverse datasets. The **labeled OpenPhish 5k dataset** served as the source of real-world phishing examples, while the **Tranco 5k dataset**, a ranked list of the world's top websites by traffic, provided high-quality benign samples. This combination ensured a robust and realistic testing environment. We evaluated two versions of our agent, one powered by the gpt-3.5-turbo model and the other by the more advanced gpt-4-turbo model, to assess the impact of the LLM's reasoning capabilities on performance.

B. Experiment 1: Brand Recognition Efficacy

The first experiment was designed to measure the agent's core competency: accurately identifying the brand being impersonated on a webpage. This is a critical prerequisite for any reference-based detection system. We randomly selected a test set of **200 phishing samples** from the OpenPhish dataset for this evaluation.

The results, summarized in TABLE I, clearly demonstrate the superior performance of the agent powered by GPT-4-turbo.

TABLE I BRAND RECOGNITION RESULTS ON 200 PHISHING SAMPLES

Detector	Correct	Wrong	Unknown
Agent-gpt-3.5-turbo	182 (91%)	5 (2.5%)	13 (6.5%)
Agent-gpt-4-turbo	190 (95%)	4 (2%)	6 (3%)

The marked improvement in correct brand recognition (from 91% to 95%) and the reduction in "Unknown" cases by more than half can be directly attributed to GPT-4's more advanced reasoning abilities and its training on a broader, more recent corpus of information. This allows it to handle more complex and ambiguous inputs with greater confidence.

Furthermore, an ablation study confirmed the value of our multi-modal preprocessing pipeline. The accuracy of using the Google

Logo Detector in isolation was found to be approximately 70%, heavily dependent on the quality of the initial logo crop. By supplementing this with GPT-4V's analysis of both the logo and the full screenshot, the system gained a more robust understanding of the visual content, leveraging GPT-4V's powerful OCR and extensive knowledge base to overcome cases where the logo was ambiguous or poorly cropped.

C. Experiment 2: End-to-End Phishing Classification Performance

The second experiment evaluated the entire integrated system, including the agent's brand prediction and the final domain checker module. We used the same 200 phishing samples from the first experiment, complemented by **200 benign samples** randomly selected from the Tranco 5k dataset, creating a balanced test set of 400 total samples.

The initial results are shown in TABLE II. The GPT-4 agent demonstrated superior performance in detecting phishing samples (True Positives), while the GPT-3.5 agent showed a slight edge in correctly identifying benign samples (True Negatives). This is because the rate of False Negatives is closely tied to the agent's failure to detect a brand, a scenario where the more capable GPT-4 agent naturally performed better.

TABLE II INITIAL PHISHING CLASSIFICATION RESULTS (400 SAMPLES)

Detector	TP	TN	FP	FN
Agent-gpt-3.5-turbo	187	186	14	13
Agent-gpt-4-turbo	194	181	19	6

However, the rate of False Positives is influenced by the domain checker. In cases where the agent correctly identifies a brand on a benign page, but that page's domain is a variant not present in the initial reference list (e.g., a blog hosted on a subdomain), a false positive occurs. To mitigate this, we conducted an optimization experiment on the domain checker, testing configurations with and without redirection checks and varying the length of the domain reference list.

TABLE III FALSE POSITIVE COUNT (OUT OF 200 BENIGN SAMPLES) UNDER DIFFERENT DOMAIN CHECKER CONFIGURATIONS

Detector	Single Domain	Single w/ Redirect	List (5) w/ Redirect	List (5)	List (10)
Agent-gpt-3.5-turbo	34	20	13	14	**12**
Agent-gpt-4-turbo	31	30	18	19	**16**

The results in TABLE III clearly show that extending the length of the domain reference list from a single domain to a list of 10 provides the most significant reduction in False Positives. This effectively accounts for the common use of domain variants by large organizations. Based on this, we adopted the **Domain List (10)** configuration without redirection checking as our final, optimized implementation.

VI. DISCUSSION AND COMPARATIVE ANALYSIS

The experimental results validate the efficacy of the ADEPT framework. This section provides a comprehensive discussion of the findings, including a direct comparative analysis against relevant baseline models to contextualize the performance of our agent-based approach.

A. Comparative Analysis Against Baseline Models

To benchmark our proposed approach, we compared its performance against two baselines that employ similar methodologies.

1. *Baseline 1: DynaPhish*: We configured the DynaPhish system with an empty knowledge base and disabled its web interaction features to create a fair comparison focused on dynamic brand identification. We evaluated it on a larger dataset of 5,000 phishing and 5,000 benign samples to ensure statistical significance. The results are presented in TABLE IV.

TABLE IV PERFORMANCE OF DYNAPHISH (WITH EMPTY KNOWLEDGE BASE)

Detector	TP (Recall)	TN (Specificity)	FP	FN
DynaPhish	1808/5000 (0.36)	4761/5000 (0.95)	239/5000 (0.05)	3191/5000 (0.64)

The results indicate that while DynaPhish is highly precise when it does make a detection (Precision of 0.88), its recall is exceptionally low (0.36). This highlights its profound dependency on a meticulously curated knowledge base. Without it, the system struggles to identify target brands, leading to a massive number of false negatives and an overall accuracy of just under 50%.

2. *Baseline 2: NLP Oneshot Prediction*: We also compared our approach to the KnowPhish system, which uses an LLM for phishing detection but employs a simpler NLP Oneshot prediction method focused only on HTML files [10]. We replicated this method by feeding processed HTML to the GPT-3.5-turbo model to predict the brand, which was then used with our domain checker.

TABLE V BRAND RECOGNITION RESULT OF NLP ONESHOT METHOD

Detector	Correct	Wrong	Unknown
NLP Oneshot (GPT-3.5)	160 (80%)	13 (6.5%)	27 (13.5%)

The rate of accurate brand prediction using only HTML content (160/200 correct) is substantially lower than that achieved by our ADEPT agent (182/200 for GPT-3.5, 190/200 for GPT-4). This discrepancy strongly suggests that the visual information captured from screenshots, which is ignored by the NLP-only method, plays a critical role and significantly enhances the accuracy of phishing detection.

B. Overall Performance and Discussion

In summary, our autonomous agent-based approach marks a notable and statistically significant advancement over existing reference-based solutions. By capitalizing on the strong language processing, understanding, and reasoning capabilities of LLMs, our framework can efficiently

and accurately harness the multi-modal information available on webpages. The final, optimized performance of our approach is presented in TABLE VI.

TABLE VI OVERALL PERFORMANCE METRICS OF THE ADEPT FRAMEWORK

Detector	Precision	Recall	Accuracy	F1-Score
agent-gpt-3.5-turbo	0.9397	0.935	0.9375	0.9373
agent-gpt-4-turbo	0.9238	0.970	**0.9450**	0.9463
DynaPhish	0.8832	0.3616	0.4999	0.5131

Unlike DynaPhish, our method does not depend on a static knowledge base, yet it demonstrates vastly superior performance, achieving an accuracy of **0.945** with the GPT-4 agent compared to DynaPhish's 0.499. This is a direct result of replacing brittle, hard-coded logic with flexible, intelligent reasoning. Additionally, by accommodating multi-modal information (text and visuals) and possessing a dynamic capability for knowledge expansion through its agentic toolkit, our approach significantly outperforms simpler LLM-based methods like the NLP oneshot prediction.

VII. LIMITATIONS AND FUTURE WORK

Despite the promising results, it is imperative to acknowledge the limitations of the current approach, which present challenges for its practical, large-scale deployment, and to outline promising directions for future inquiry.

A. Limitations of the Study

One of the most significant limitations is the **high runtime cost**. The operational latency is

largely dependent on the number of API calls the agent makes to its toolkit (Google Search, GPT-4V, etc.). The average runtime for analyzing a single sample was measured to be **approximately 20 seconds**. In a real-time threat detection scenario, such as an email gateway processing thousands of URLs per minute, this duration is computationally expensive and presents a major scalability challenge.

Another limitation is the **constrained number of interaction rounds**. The agent is currently restricted to a maximum of five tool-use cycles to manage runtime costs. While sufficient for most cases, certain complex webpages may require additional investigation to definitively resolve brand ambiguity. This fixed limit on the agent's reasoning process could potentially impact prediction accuracy in edge cases.

B. Future Work and Possible Improvements

Based on the identified limitations, we propose several avenues for future research:

Optimization of Agent Interaction: Future work should focus on determining the optimal number of agent-tool interactions. This could involve dynamically ending the investigation once the agent reaches a high confidence score, rather than using a fixed number of cycles, thereby finding a better balance between accuracy and efficiency.

Integration of a Dynamic Knowledge Base (Cache): A significant improvement would be to integrate the agent with a dynamic knowledge base that functions as a high-speed cache. This knowledge base would be populated *solely* from the analysis of benign samples. When the agent confidently classifies a webpage as benign, its domain and brand identity would be added to this trusted cache. For subsequent encounters with the same domain, the system could rely on the cached result, bypassing the costly agent-based analysis entirely. This would dramatically reduce runtime for frequently visited legitimate sites while ensuring the integrity of the knowledge base.

Cost-Benefit Analysis of Different LLMs: The current study compared GPT-3.5 and GPT-4. Future work could include a broader analysis of other available LLMs, including smaller, open-source models. This would enable a cost-benefit analysis to determine if a less costly model could achieve a comparable level of accuracy, potentially making the solution more economically viable for widespread deployment.

VIII. CONCLUSION

This research has successfully designed, implemented, and validated a novel, reference-based framework for automated phishing detection that leverages the power of an autonomous LLM agent. The study began with a critical analysis of the architectural flaws in existing static and dynamic detection systems, proposing an agent-based alternative that mimics human analytical processes to overcome their inherent brittleness.

The primary contributions of this work are threefold. First, it introduces a new paradigm for phishing detection that effectively addresses the scalability and adaptability challenges of systems reliant on static knowledge bases. Second, it demonstrates empirically that an LLM-powered agent,

equipped with a toolkit for multi-modal analysis and real-time information retrieval, can achieve a state-of-the-art level of performance in both brand recognition and overall phishing classification. Finally, it provides a comprehensive blueprint and a proof-of-concept implementation that serves as a foundation for future research into the application of autonomous AI agents in the broader field of cybersecurity.

While the current implementation has notable limitations in terms of runtime cost, the findings of this paper strongly indicate that the agent-based approach represents a significant and promising advancement in the ongoing effort to combat the pervasive and evolving threat of phishing.

REFERENCES

[1] Raihan, M. Fadhli, and L. Lindawati, "Implementation of deep learning for detecting phishing attacks on websites with combination of CNN and LSTM," Jurnal Teknik Informatika (Jutif), vol. 11, no. 5, pp. 2446–2456, 2024.

[2] S. Younus, S. Almoshity, and A. Moftah, "Survey of website phishing detection based deep learning approach," African Journal of Advanced Pure and Applied Sciences (AJAPAS), vol. 6, no. 5, pp. 478–488, 2024.

[3] A. A. Albishri and M. M. Dessouky, "A comparative analysis of machine learning techniques for URL phishing detection," Engineering, Technology & Applied Science Research, vol. 14, no. 6, pp. 18495–18501, Dec. 2024.

[4] S. Aslam, H. Aslam, A. Manzoor, C. Hui, and A. Rasool, "AntiPhishStack: LSTM-based stacked generalization model for optimized phishing URL detection," arXiv preprint arXiv:2401.08947, Jan. 2024.

[5] M. S. Islam Ovi, M. H. Rahman, and M. A. Hossain, "PhishGuard: A multi-layered ensemble model for optimal phishing website detection," arXiv preprint arXiv:2409.19825, Sep. 2024.

[6] A. Fajar, S. Yazid, and I. Budi, "Enhancing phishing detection through feature importance analysis and explainable AI: A comparative study of CatBoost, XGBoost, and EBM models," arXiv preprint arXiv:2411.06860, Nov. 2024.

[7] T. Ige, C. Kiekintveld, A. Piplai, A. Waggler, O. Kolade, and B. H. Matti, "An investigation into the performances of the current state-of-the-art Naive Bayes, non-Bayesian and deep learning based classifier for phishing detection: A survey," arXiv preprint arXiv:2411.16751, Nov. 2024.

[8] D. Prasad, "Threat Spotlight: A million phishing-as-a-service attacks in two months highlight a fast-evolving threat," Barracuda Networks, Mar. 2025.

[9] M. J. K. O'Neill, "Most people still can't identify a phishing attack written by AI - and that's a huge problem, survey warns," TechRadar, Oct. 2025.

[10] J. Smith, "Most adults couldn't differentiate between authentic and AI phishing emails, new survey shows," New York Post, Oct. 2025.

[11] Microsoft, "Microsoft blocks phishing scam which used AI-generated code to trick users," TechRadar, Oct. 2025.

[12] A. Jadhav and P. Chandre, "A hybrid heuristic-machine learning framework for phishing detection using multi-domain feature analysis," Engineering, Technology & Applied Science Research, vol. 15, no. 5, pp. 27219–27226, Oct. 2025.

[13] D. Gada, "Indian phishing landscape: A machine learning and deep learning approach for detecting malicious URLs and curating an indigenous dataset," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 4, pp. 1670–1679, Jun. 2024.

[14] M. W. Shaukat, R. Amin, M. M. A. Muslam, A. H. Alshehri, and J. Xie, "A hybrid approach for alluring ads phishing attack detection using machine learning," Sensors, vol. 23, no. 19, pp. 8070, Sep. 2023.

[15] G. Mohamed, J. Visumathi, M. Mahdal, J. Anand, and M. Elangovan, "An effective and secure mechanism for phishing attacks using a machine learning approach," Processes, vol. 10, no. 7, pp. 1356, Jul. 2022.

[16] T. Guo, X. Chen, Y. Wang, R. Chang, S. Pei, N. V. Chawla & O. Wiest, "Large Language Model Based Multi-agents: A Survey of Progress and Challenges," in Proc. 33rd International Joint Conference on Artificial Intelligence (IJCAI 2024 – Survey Track), pp. 8048–8057, 2024.

[17] L. Wang, C. Ma, X. Feng et al., "A Survey on Large Language Model Based Autonomous Agents," Frontiers of Computer Science, vol. 18, article 186345, 2024.

[18] "Digital Payment Fraud in High Growth Markets Study from LexisNexis Risk Solutions Finds 90% of Respondents Experienced an Increase in Online Fraud Over Past 12 Months," LexisNexis Risk Solutions (Press Release), 2022.

[19] "New report reveals an increase in digital payment fraud in Europe," Tietoevry Banking Insight Report, 30 April 2025.

[20] "A Survey of Machine Learning-Based Solutions for Phishing Website Detection," MDPI conference/journal article, 2023.

Conflict of Interest Statement: *The authors declare that there is no conflict of interest regarding the publication of this paper.*

Generative AI Statement: *The author confirms that no Generative AI tools were used in the preparation or writing of this article.*

Publishers Note: *All statements made in this article are the sole responsibility of the author(s) and do not necessarily reflect the views of their affiliated institutions, the publisher, editors, or reviewers. Any products mentioned or claims made by manufacturers are not guaranteed or endorsed by the publisher.*

Cite this Article

Ashwarya Singh, Kalpana Mishra. ADEPT: An Autonomous, LLM-Powered Agent for Dynamic, Reference-Based Phishing Detection. International Research Journal of Engineering & Applied Sciences (IRJEAS). 12(4), pp. 81-96, 2025. 10.55083/irjeas.2025.v13i04009.