Do We Really Need Reference-based Phishing Detection? Unleashing the Power of GNN

Tailai Song*, Pedro Casas†, Michela Meo*
*Politecnico di Torino, Turin, Italy, †AIT Austrian Institute of Technology, Vienna, Austria
*first.last@polito.it, †pedro.casas@ait.ac.at

***Abstract*—Reference-based approaches have risen to prominence in phishing detection, attributed to their aptitude for extracting brand intention, performing comparisons with established references, and utilizing advanced tools. Despite their notable success, such methods are typically underpinned by highly specialized frameworks that are sophisticated yet burdensome in terms of scale, technological requirements, and cost, factors that substantially hinder their practicality, scalability, and maintainability. In this paper, we revisit the necessity of reference-based techniques and contend that the under-performance of conventional machine learning (ML) algorithms stems not from the absence of references, but from inherent model limitations. To this end, we propose a graph neural network (GNN)-based approach that capitalizes on the intrinsic properties of webpages, the domain and the document object model (DOM) tree hierarchy. We formalize a binary classification problem, and our model achieves competitive performance without relying on intricate architectures or unattainable resources/infrastructures, in contrast to reference-based counterparts.**

***Index Terms*—Phishing detection, cybersecurity, machine learning, deep learning, graph neural network**

I. Introduction

Phishing attacks, a notorious form of social engineering that disguises as legitimate webpages to deceive users into revealing sensitive information, constitute one of the most detrimental cyber threats, often resulting in substantial financial losses. To cope with this challenge, extensive efforts have been devoted to phishing detection, including both heuristic and artificial intelligence (AI)-driven solutions [1], [2]. They typically employ periodically updated blacklists [3] or supervised ML/deep learning (DL) algorithms, wherein features are predominantly derived from URLs and HTML content [4], [5].

In recent years, reference-based approaches have exhibited superior performance and become state-of-the-art (SOTA) solutions [6]–[8]. Specifically, they leverage comprehensive resources and tools to infer a webpage's brand intention by extracting its brand representation (e.g., logo or screenshot) and comparing it against a reference list comprising legitimate websites with auxiliary information such as authentic domains. However, there is no free lunch—enhanced effectiveness inevitably incurs a commensurate cost. Due to the demands of multidimensional knowledge retrieval, intricate brand comparison, continuous reference upkeep, etc., reference-based systems commonly necessitate complicated and costly technologies under the hood, such as large language models (LLMs) and search engines. In consequence, such solutions possessed by big players, although promising, remain unaffordable to actual victims, the consumers.

In this context, we rethink the purported indispensability of reference-based phishing detection, positing that the architectural inadequacies, the superficiality of feature design, and the flawed performance evaluation underpin the deficiency of alternative models. Accordingly, we propose a novel graph neural network (GNN)-based framework, augmented by meticulously-crafted webpage features, the HTML DOM tree structure, and node-specific message passing. We aim to distinguish between benign and phishing websites, considering two distinct datasets and comparing against multiple ML/DL-based and reference-based approaches. Preliminary results underscore the prospect of utilizing GNNs to effectively identify malicious webpages in an end-to-end manner devoid of the complexity associated with reference-based methods.

II. Problem context

Phishing comprises a series of fraudulent activities in which attackers manipulate individuals into disclosing private information or inadvertently installing malware or ransomware. It is nowadays the most prevalent form of cybercrime, surpassing all other categories in frequency of reported incidents [9]. Typically, adversaries leverage the so-called phishing kit, a set of preassembled, ready-to-deploy packages designed to facilitate the creation of counterfeit webpages and the execution of phishing campaigns, to masquerade as real websites with the objective of stealing confidential data (e.g., accounts and passwords) from users. Phishing detection is normally formulated as a binary classification task, where various pertinent elements of a webpage including the URL, HTML, HTTP request/response, and screenshot can be used as input to uncover its label, either benign or phishing.

Conventionally, phishing detection can be classified into three categories: heuristics that examine webpage attributes such as HTML structure and suspicious keywords [10], [11], blacklists/whitelists that validate unknown URLs against established repositories [12]–[14], and rule-based systems that adopt predefined rules based on specific phishing patterns [15], [16]. Nonetheless, they fall short in adapting to emerging phishing technologies, primarily owing to their reliance on static, historical, and inflexible criteria.

ML and DL algorithms have emerged as effective candidates for phishing detection [17]–[19]. They usually leverage

978-3-903176-74-4

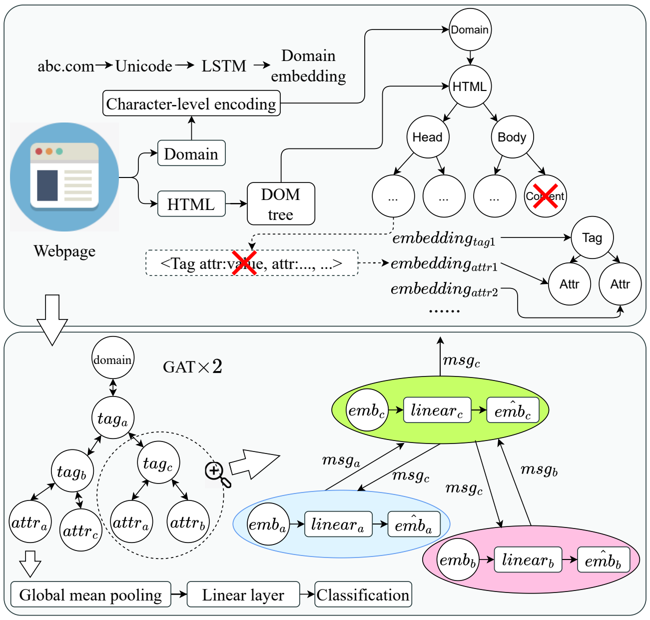

Fig. 1: Framework overview.

features extracted from URLs and HTML to enable automated classifications via either traditional ML models [20], [21] or advanced DL techniques [4], [22], [23]. GNNs have also garnered increasing attention in modeling HTML structures/contents [24]–[26] or correlating external resources [27], [28]. Nevertheless, the majority of these works fundamentally suffer from the encoding of web content, as the underlying models are constructed upon a fixed corpus and thus struggle to generalize to new phishing websites.

Reference-based approaches represent a broad array of solutions, such as visual- and NLP-based technologies that extract multidimensional information from webpages to evaluate their fidelity against known benign exemplars [29]. They typically construct a brand knowledge base with logos, aliases, and legitimate domains, which serves as a reference for a detector backbone in the phishing detection pipeline [8], [30]–[34]. Although showcasing prowess, the sophistication behind inherently imposes substantial constraints not only on end users but also on the research community—from inaccessible tools (e.g., LLMs) to prohibitive cost (e.g., API subscriptions) and from complicated implementation (e.g., multimodal architectures) to unmatchable stakeholders (e.g., large companies). Collectively, these factors essentially undermine the practicality, reproducibility, and reliability of reference-based solutions.

III. Methodology

We introduce a GNN-based framework which exploits the domain and DOM structure of webpages and incorporates tailored message passing, as portrayed in Figure 1. To facilitate a lightweight and streamlined phishing detection, we refrain from considering intricate content, focusing instead on extracting readily accessible information, specifically the domain and HTML.

A. Feature encoding

The initial step involves encoding various elements in a webpage. For the domain, we resort to character-level encoding, where each character is mapped to its corresponding Unicode value sequentially, yielding a list of numerical representations. We refer to Unicode as it has universal coverage of any character, circumventing the out-of-vocabulary issue. Afterwards, the domain list is passed through an LSTM (Long Short-Term Memory) layer to produce a latent domain representation, which is then subjected to mean pooling along the feature dimension to derive the final domain embedding.

As for the HTML composed of the DOM tree structure, the process is rather complex. Each element in the tree consists of a tag and one or more attribute-value pairs, while the leaf nodes contain webpage contents. First, we exclude values and contents for two main reasons: 1) They can constitute arbitrary words, characters, digits, or punctuation in any order or quantity, leading to a tokenization scale akin to general NLP problems. Besides being infeasible and impractical, this approach would hinder the model's ability to generalize over novel malicious websites if only the corpus of existing datasets is considered. 2) Since phishing webpages intend to imitate legitimate ones, the content eventually displayed to users (rather than embedded in the HTML structure) is suppose to resemble that of authentic websites, rendering it uninformative and even misleading. Concerning tags and attributes, a fixed corpus exists, comprising up to hundreds of distinct entities as defined by the HTML protocol. Moreover, we treat attributes as sub-nodes for a tag node to construct a nested sub-tree. In this context, we create and initialize an equivalent amount of trainable parameters, learning and refining node embeddings during the training process. Additionally, we supplement two more embeddings to deal with unknown entities when encountering modern customized tags and attributes, which are much less frequent than standard ones.

At last, we integrate the domain embedding into the DOM tree by linking it to the HTML root node, thereby forming the final webpage graph. Generally, we aim to explore and elaborate on the HTML protocol, uncovering the structural, low-level differences between benign and malicious samples, as phishing webpages are often developed atop a set of unique phishing kits, which provide particular functionalities rarely observed in normal websites.

B. GNN

The DOM tree is a special natural graph structure, warranting compatibility with GNN to investigate the underlying correlations within the HTML structure. We employ Graph Attention Networks (GAT) [35] as the base model and incorporate custom node-specific layers to improve performance. Unlike many cases involving homogeneous graphs, our DOM graph is heterogeneous, with each node belonging to a certain tag or attribute. In general, GNNs including GAT utilize a single unified linear layer to update node features and determine the message exchanged, which lacks the freedom of accommodating diverse nodes. In our case, we categorize all available tags and attributes into 13 classes[1], with each category assigned to a dedicated linear layer. As a result, for a node of category c, the updated features (i.e., messages) are computed using the corresponding linear layer, $linear_c$. Overall, we apply two layers of GAT with these customized linear layers, whose output is fed into global mean pooling to generate the final graph representation. Finally, we employ an additional linear layer for binary classification.

IV. Experiment

A. Experiment setup

To evaluate the model generalization, we employ two distinct datasets collected in 2021 [36] and 2023 [31], wherein the former one includes 29,496 phishing webpages using OpenPhish Premium Service [12] and 29,951 benign webpages sourced from the top-ranked Alexa list [37], and the latter one contains manually labeled and balanced collection with 5,000 benign samples randomly selected from Tranco [38] and 5,000 phishing samples obtained from OpenPhish [12].

To benchmark our approach, we compare it with three baseline ML/DL models: HTMLPhish [22], HTML-GNN [24], and Web2Vec [23]. On top of that, we consider two SOTA reference-based methods: PhishPedia [36] and KnowPhish [31]. We conduct two trials of model development, utilizing one dataset for training and the other one for evaluation, by assessing standard classification metrics including accuracy, recall, and precision. Notably, because of the heightened requirements, we are unable to reproduce the outcomes of the two reference-based models, and therefore, we quote the results from their original publications for comparison and do not evaluate performance on alternative datasets.

B. Experimental result

Table I summarizes the experimental results for each model and dataset. At first glance, the two ML models, HTMLPhish and Web2Vec completely fail the tasks, with classification accuracies hovering around 60% for both datasets. The deficiency originates from their ignorance on domain shift, as both models extract a fixed semantic corpus from their respective datasets[2]. HTML-GNN, which leverages GNN and is cautious with feature encoding, delivers moderate yet insufficient performance. While our model achieves substantial improvements over the baselines, reference-based approaches still dominate the board with superior performance especially when it comes to precision. In particular, our model yields comparable recalls (even surpassing PhishPedia on dataset 2), but trails in precision, signifying a non-ignorable number of false positives. Despite this, we remain optimistic about the feasibility of superseding reference-based approaches and affirm the promise of GNNs primarily for three reasons: 1) This work is still in progress, and while the current model is not fully optimized, its preliminary performance already demonstrates strong potential for future enhancement. 2) The ostensibly enhanced results of reference-based models, drawn from separate experimental settings, are not directly comparable with ours, complicating a definitive evaluation. 3) While reference-based models may remain dominant, the balance between complexity and effectiveness highlights the practical value of our approach, which offers a cost-effective and accessible solution for phishing detection.

TABLE I: Experimental results.

	Model	Accuracy	Recall	Precision
Dataset 1	HTMLPhish	52.0	52.0	60.3
	Web2Vec	57.2	57.2	72.2
	HTML-GNN	84.7	84.7	84.8
	PhishPedia	–	–	–
	KnowPhish	92.5	86.9	97.8
	Ours	85.9	85.9	86.6
Dataset 2	HTMLPhish	57.6	61.4	66.1
	Web2Vec	64.9	59.7	78.1
	HTML-GNN	83.3	80.9	88.0
	PhishPedia	99.2	87.1	98.2
	KnowPhish	–	–	–
	Ours	91.0	90.8	90.7

V. Conclusion

In this work, we investigate the strengths and limitations of various phishing detection methodologies, in which heuristic and ML/DL-based solutions fall short in promptness and generalization, while SOTA reference-based ones, albeit effective, impose restrictive demands that hinder widespread adoption. To address these challenges, we propose an end-to-end, lightweight, and cost-effective phishing detector based on GNN. We leverage the readily accessible webpage content—the domain and HTML structure, extracting informative features while discarding noisy, meaningless elements. We encode domain names at the character level and exploit the graph structure inherent in the DOM tree. Furthermore, we enhance GAT with customized node-specific layers to improve model adaptability and performance. The experiment is undertaken involving two independent datasets and multiple contrasts encompassing DL models and reference-based approaches. Conclusively, our model significantly outperforms baselines and obtains acceptable performance relative to reference-based solutions. Future work could consist of more rigorous model development and evaluation to further enhance generalization and robustness.

Acknowledgment

This work has been supported by the FWF Austrian Science Fund, Project reference I-6653, as part of the EU CHIST-ERA-2022-SPiDDS-02 project *GRAPHS4SEC*. The work has been also partially supported by Cisco Systems Inc., the European Union under the Italian National Recovery and Resilience Plan (NRRP) of NextGenerationEU, partnership on “Telecommunications of the Future” (PE00000001 - program “RESTART”, Focused Project R4R), and the SmartData@PoliTO center on Big Data and Data Science.

[1]For example, `body`, `p`, and `title` belong to the types of sectioning, text, and document, respectively.

[2]These models demonstrate exceptional performance when trained and tested on the same dataset using a conventional split, but such results reflect overfitting and poor generalization.

References

[1] N. Q. Do, A. Selamat, O. Krejcar, E. Herrera-Viedma, and H. Fujita, “Deep learning for phishing detection: Taxonomy, current challenges and future directions,” *Ieee Access*, vol. 10, pp. 36429–36463, 2022.

[2] M. Alanezi, “Phishing detection methods: A review.,” *Technium*, vol. 3, no. 9, 2021.

[3] J. Bayer, S. Maroofi, O. Hureau, A. Duda, and M. Korczynski, “Building a resilient domain whitelist to enhance phishing blacklist accuracy,” in *2023 APWG Symposium on Electronic Crime Research (eCrime)*, pp. 1–14, IEEE, 2023.

[4] H. Le, Q. Pham, D. Sahoo, and S. C. H. Hoi, “Urlnet: Learning a url representation with deep learning for malicious url detection,” 2018.

[5] S. Asiri, Y. Xiao, S. Alzahrani, S. Li, and T. Li, “A survey of intelligent detection designs of html url phishing attacks,” *IEEE Access*, vol. 11, pp. 6421–6443, 2023.

[6] S. Abdelnabi, K. Krombholz, and M. Fritz, “Visualphishnet: Zero-day phishing website detection by visual similarity,” 2020.

[7] S. Afroz and R. Greenstadt, “Phishzoo: Detecting phishing websites by looking at them,” pp. 368 – 375, 10 2011.

[8] R. Liu, Y. Lin, X. Yang, S. H. Ng, D. M. Divakaran, and J. S. Dong, “Inferring phishing intention via webpage appearance and dynamics: A deep vision based approach,” in *31st USENIX Security Symposium (USENIX Security 22)*, pp. 1633–1650, 2022.

[9] Federal Bureau of Investigation, “Internet Crime Report 2020,” March 2021. FBI Internet Crime Complaint Center. Retrieved 21 March 2021.

[10] L. A. T. Nguyen, B. L. To, H. K. Nguyen, and M. H. Nguyen, “A novel approach for phishing detection using url-based heuristic,” in *2014 international conference on computing, management and telecommunications (ComManTel)*, pp. 298–303, IEEE, 2014.

[11] R. S. Rao, A. R. Pais, and P. Anand, “A heuristic technique to detect phishing websites using twsvm classifier,” *Neural Computing and Applications*, vol. 33, no. 11, pp. 5733–5752, 2021.

[12] OpenPhish, “Openphish - phishing intelligence feed,” 2025. Accessed: 15 April 2025.

[13] PhishTank, “Phishtank - join the fight against phishing,” 2025. Accessed: 15 April 2025.

[14] N. A. Azeez, S. Misra, I. A. Margaret, L. Fernandez-Sanz, and S. M. Abdulhamid, “Adopting automated whitelist approach for detecting phishing attacks,” *Computers & Security*, vol. 108, p. 102328, 2021.

[15] Y. Mourtaji, M. Bouhorma, D. Alghazzawi, G. Aldabbagh, and A. Alghamdi, “Hybrid rule-based solution for phishing url detection using convolutional neural network,” *Wireless Communications and Mobile Computing*, vol. 2021, no. 1, p. 8241104, 2021.

[16] M. SatheeshKumar, K. Srinivasagan, and G. UnniKrishnan, “A lightweight and proactive rule-based incremental construction approach to detect phishing scam,” *Information Technology and Management*, vol. 23, no. 4, pp. 271–298, 2022.

[17] C. Catal, G. Giray, B. Tekinerdogan, S. Kumar, and S. Shukla, “Applications of deep learning for phishing detection: a systematic literature review,” *Knowledge and Information Systems*, vol. 64, no. 6, pp. 1457–1500, 2022.

[18] R. Zieni, L. Massari, and M. C. Calzarossa, “Phishing or not phishing? a survey on the detection of phishing websites,” *IEEE Access*, vol. 11, pp. 18499–18519, 2023.

[19] A. Basit, M. Zafar, X. Liu, A. R. Javed, Z. Jalil, and K. Kifayat, “A comprehensive survey of ai-enabled phishing attacks detection techniques,” *Telecommunication Systems*, vol. 76, pp. 139–154, 2021.

[20] Y. Li, Z. Yang, X. Chen, H. Yuan, and W. Liu, “A stacking model using url and html features for phishing webpage detection,” *Future Gener. Comput. Syst.*, vol. 94, pp. 27–39, 2019.

[21] S. Venugopal, S. Y. Panale, M. Agarwal, R. Kashyap, and U. Ananthanagu, “Detection of malicious urls through an ensemble of machine learning techniques,” in *2021 IEEE Asia-Pacific Conference on Computer Science and Data Engineering (CSDE)*, pp. 1–6, IEEE, 2021.

[22] C. Opara, B. Wei, and Y. Chen, “Htmlphish: Enabling phishing web page detection by applying deep learning techniques on html analysis,” in *2020 International Joint Conference on Neural Networks (IJCNN)*, p. 1–8, IEEE, July 2020.

[23] J. Feng, L. Zou, O. Ye, and J. Han, “Web2vec: Phishing webpage detection method based on multidimensional features driven by deep learning,” *IEEE Access*, vol. 8, pp. 221214–221224, 01 2020.

[24] L. Ouyang and Y. Zhang, “Phishing web page detection with html-level graph neural network,” in *2021 IEEE 20th international conference on trust, security and privacy in computing and communications (trustCom)*, pp. 952–958, IEEE, 2021.

[25] J. Lindamulage, L. MandiraPabasari, S. Yapa, I. Perera, and J. Krishara, “Vision gnn based phishing website detection,” in *2023 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, pp. 1–7, IEEE, 2023.

[26] C. Wang, Z. Liu, and Y. Zeng, “Detecting phishing urls with gnn-based network inference,” in *2024 International Conference on Networking and Network Applications (NaNA)*, pp. 504–509, IEEE, 2024.

[27] T. Bilot, G. Geis, and B. Hammi, “Phishgnn: A phishing website detection framework using graph neural networks,” 07 2022.

[28] T. Kim, N. Park, J. Hong, and S.-W. Kim, “Phishing url detection: A network-based approach robust to evasion,” in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1769–1782, 2022.

[29] W. Li, S. Manickam, Y.-W. Chong, W. Leng, and P. Nanda, “A state-of-the-art review on phishing website detection techniques,” *IEEE Access*, 2024.

[30] R. Liu, Y. Lin, Y. Zhang, P. H. Lee, and J. S. Dong, “Knowledge expansion and counterfactual interaction for {Reference-Based} phishing detection,” in *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 4139–4156, 2023.

[31] Y. Li, C. Huang, S. Deng, M. L. Lock, T. Cao, N. Oo, H. W. Lim, and B. Hooi, “{KnowPhish}: Large language models meet multimodal knowledge graphs for enhancing {Reference-Based} phishing detection,” in *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 793–810, 2024.

[32] R. Liu, Y. Lin, X. Teoh, G. Liu, Z. Huang, and J. S. Dong, “Less defined knowledge and more true alarms: Reference-based phishing detection without a pre-defined reference list,” in *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 523–540, 2024.

[33] J. Lee, P. Lim, B. Hooi, and D. M. Divakaran, “Multimodal large language models for phishing webpage detection and identification,” *arXiv preprint arXiv:2408.05941*, 2024.

[34] T. Cao, C. Huang, Y. Li, W. Huilin, A. He, N. Oo, and B. Hooi, “Phishagent: a robust multimodal agent for phishing webpage detection,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 39, pp. 27869–27877, 2025.

[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” 2018.

[36] Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, F. Zhang, and J. S. Dong, “Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages,” in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 3793–3810, 2021.

[37] Alexa Internet, Inc., “Alexa site info,” 2022. Accessed: 15 April 2025.

[38] V. L. Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, “Tranco: A research-oriented top sites ranking hardened against manipulation,” *arXiv preprint arXiv:1806.01156*, 2018.