PhishEye: A Dual-Layer AI-Powered Phishing Detection System

Waleed AlGobi[1], Atika Alnaim[2], Fatimah Almusaid[3] , Rawan Alali[4]
King Fahd University of Petroleum and Mineral
The Department of Information and Computer Science
G202306850@kfupm.edu.sa, G202306890@ kfupm.edu.sa, G202307110@ kfupm.edu.sa

Submitted in partial fulfillment of the requirements for the degree of Master of Cybersecurity

The Department of Information and Computer Science
King Fahd University of Petroleum and Mineral (KFUPM)
Dhahran, Saudi Arabia
April 12, 2025

PhishEye: A Dual-Layer AI-Powered Phishing Detection System

Waleed AlGobi[1], Atika Alnaim[2], Fatimah Almusaid[3] , Rawan Alali[4]
King Fahd University of Petroleum and Mineral
The Department of Information and Computer Science
G202306850@kfupm.edu.sa, G202306890@ kfupm.edu.sa, G202307110@ kfupm.edu.sa

Abstract

PhishEye is an AI-powered phishing detection system which works by combining URL classification with Phishpedia's logo-based verification. PhishEye processes by first checking the domain-brand consistency structure under which it searches for logos and applies a Transformer-based model if the logo is sensed to be missing or unrecognized. This is done through a two-layer detection strategy which is highly effective due to its feature of early detection [1]. The early detections play important role as it significantly reduces the number of false positives and helps the model in identifying phishing sites who target less well-known brands or companies.

1. Introduction

Phishing is considered to be among the most significant cybersecurity threats nowadays as the attacker can create fake websites that mimic real services with the sole purpose of stealing personal data. Therefore, various existing detection methods, like blacklists and heuristic-based systems are required to be constantly updated for maintaining the dynamic nature of phishing campaigns. Even in these phishing attacks, the dynamic nature of attacks renders various machine learning analyses to be avoided easily. For a long time, Phishpedia had been used which mainly made use of a consistency-based approach. This process of this model mainly included detecting phishing by analysing webpage structure.

As a replacement for Phishpedia, the PhishEye could be used as this model uses a two-layered phishing detection system which also relies on logo-based verification to detect fake brand impersonations and potential Phishing attacks. Within this model, the Transformer-based approach is also used whihc analyses URL structures and content for phishing indicators [2]. Thus, the proposed model can be deemed effective in enhancing phishing detection as it processes by combining Phishpedia's scalability and accuracy with recent AI-based URL analysis advancements. Besides, our key contributions to it are:

- Within this model, we introduced a second detection layer. The second detection layers is mainly used for analysing URL patterns when brand logos are not found or mis-matched.
- We made certain that PhishEye has high effeciecny in switching between visual logo verification and AI-based URL analysis. We also ensured that model must only switich between logo verification and url analysis depending upon the information that it looks out for.
- We also conducted a systematic evaluation of PhishEye which helped us in concluding that the model was immense effective in reducing false negatives results in comparison Phishpedia. The model also maintained higher accuracy across diverse phishing datasets.

2. Overview of PhishEye

2.1 Threat Model

The PhishEye model process is through targeting deceptive URLs and fake branding tactics which most attacks use for mimicking the authenticity of the actual brand. Previously, the phishing detections had relied extensively on matching logos and domains and they often failed when logos were either missing or the brand wasn't preloaded. This, thus, causes some inaccurate results as phishing sites were being considered safe [3]. Therefore, PhishEye can be used as it effectively uses a Transformer-based model which verifies the domain and logo for detecting potential phishing attacks. Within this model, the Transformer-based approach is structured for analyzing URLs directly, which helps the model in detecting probable phishing sites even when visual cues are unavailable or unrecognized. Thus, it can also be denoted that the use of the PhishEye model effectively improves the protection against sophisticated phishing attempts when compared with other models.

2.2 System Overview

The PhishEye model is composed of a two-layered detection system with the first layer detection system being used for verifying logos through image analysis and matching domains. In this layer, if any mismatch is discovered a phishing alert is triggered and the site is highlighted as a phishing site. In the instance where no recognizable logo is found, the second layer analysis is used in which the Transformer-based approach is further used for classifying the URL. Under this approach, The URLs are analyzed for their structure and patterns for detecting potential Phishing attack [2]. The importance of such a hybrid approach can be highlighted by its use of visual and URL analysis, which effectively renders the model scalable, explainable, and protective against phishing sites, that have been using new brands or brands which lack any apparent visual cues [3].

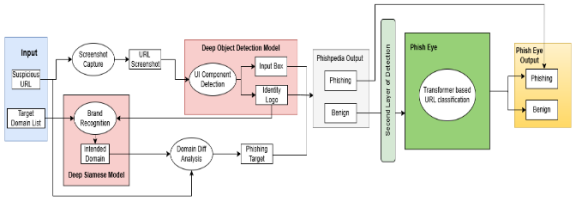

Figure-1: PhishEye Framework: a hyprid deep learning system that uses visual and URL analysis.

3. Design and Development of PhishEye

Previously, Phishpedia's version was processed by mainly matching a webpage's logo with a preloaded brand and it also looked for domain mismatch. However, with such an approach If the brand was not available in the reference set, the system would then denote the URL as benign. A prime example of such an issue can be WhatsApp; if WhatsApp isn't included, a fake WhatsApp URL would remain undetected [1]. Thus, the solution with such an approach is to expand the logo database which is possible with the constant addition of known brand logos and domains. Besides, such a solution can be denoted as effective in improving detection accuracy for a wider range of phishing URLs [2].

Primarily, with the use of Phishpedia, The system faces a huge problem in loading numerous brand logos as it hinders the system's accuracy of URL classification. The Phishpedia model also misclassifies the URLs as safe and legitimate in the instances where logo preloading fails. Such inaccuracy can also occur due to URLs being deceptive or lacking any potential feature to give an accurate prediction. Thus, to address this problem, the solution is to use the model which uses the transformer-based approach within which the additional layer helps in reevaluating the cases that are considered uncertain by the model. This model would then proceed with the textual and structural-based URL analysis as such analysis would improve the classification process in the instances where the logos are unavailable.

Mainly, as the PhishEye exists as a two-stage hybrid phishing detection framework, it is capable of improving the Phishpedia through effectively combining visual logo-domain checks. The logo check further makes use of transformer-based URL semantic analysis which effetivley covers the logo-based (e.g., fake PayPal) and text-based (e.g., typo squatted "faceb00k-login[.]com") types of phishing attack [5].

3.1. Technical Design

3.1.1. Framework

Under the framework of this model, the Hugging Face is mainly used for development and deployment. Hugging Face is an open-source platform that effectively changes the landscape of NLP through its pre-trained models like BERT and GPT [6]. These models as further offered within Hugging Face's Transformers library. It is important to highlight that such a model simplifies fine-tuning and deployment with user-friendly APIs while such tuning and deployment then enables a sense of rapid experimentation and broader access towards advanced language technologies for researchers and developers.

3.1.2. Algorithm

The imanoop7/ Bert-phishing-detector is the detector which uses the pre-trained BERT model which is fine-tuned for detecting a variety of phishing attacks through various text inputs like URLs and emails. Within this model, after preprocessing and tokenization are adequately adhered to, the text is then converted into various contextual embeddings that further feed the classification layer to effectively distinguish phishing from benign contents. It can also be perceived that this model works best through cross-entropy loss, backpropagation, and hyperparameter tuning as this helps the model in enhancing its precision and optimises its accuracy. The Dropout layers and learning rate schedulers also play an

important role in this model as they help in preventing overfitting. The use of modular design is also a major component in this model which ensures that the design is capable of specifically supporting real-time deployments which would further enhance the cybersecurity systems. It is important to denote the fact that such components are ineffective for showcasing how the power of transfer learning is adequately applicable in tackling phishing. Thus, this also opens avenues for further research throughout the domain of adaptive threat detection and responses.

3.2 Two-Stage Architecture

3.2.1. Phishpedia Legacy Pipeline

Logo Detection:

Algorithm: Within the algorithm, the Faster R-CNN (Region-based Convolutional Neural Network) is enabled by using Facebook's Detectron2 framework. Process: Input: A full-page image or screenshot of the target URL is taken first of all. Then the Object detection properly identifies various bounding boxes for various candidate logos while it uses a pre-trained ResNet-50 backbone to do so. Various non-maximum suppression (NMS) filters are used specifically to overlap boxes. Such overlapping causes the boxes to further possess only high-confidence detections. Output: The output comes out to be a Cropped logo region.

Brand Matching and URL Check:

Algorithm: under the brand matching algorithm, the Siamese Neural Network is used with triplet loss. Process: The model extracts feature embeddings for various detected logos through the use of pre-trained CNN (e.g., ResNet-34). The model then matches embeddings against a database of 181 preloaded or available brand logos (Target list). The matches are further validated through their cosine similarity > 0.92. Domain Check: under domain check, the URL would be checked and verified from the brand's legitimate domain (like in “paypal.com" vs. "paypa1-login.net").

```
Def Phisheye (URL: str, screenshot: Image) -> str:
   # Stage 1: Phishpedia
   logos = detect logos (screenshot,
confidence=0.85)
   For logo in logos:
      brand = match_logo(logo, targetlist,
similarity_threshold=0.92)
      if brand:
         if not is_legitimate_domain(URL,
brand.domains):
            return "Phishing (Logo-Domain
Mismatch)"
         else:
            return "Benign (Verified Brand)"

   # Stage 2: BERT Classifier
   prediction = URL_Classifier(URL)
   return "Phishing (BERT)" if prediction == 1 else
"Benign (BERT)"
```

Figure-2: Shows Two-Stage Phishing Detection Workflow in PhishEye.

3.2.2. Transformer-Based URL Classifier

The classifier mainly uses the algorithm of Fine-tuned BERT (Bidirectional Encoder Representations from Transformers) as algorithm to enhance its input accuracy. In term of processing, it returns "benign" (no logo match) under which it tokenizes raw URL strings that are later fed to imanoop7/Bert-phishing-detector. After this process, the model moves towards analysing various lexical patterns (e.g., suspicious substrings, length, special characters) as well as contextual semantics. As illustrated in Figure 2, The Outputs for this classifier would be mainly some binary classifications such as: "phishing" or "benign."

3.3. Design Justification

The PhishEye improves phishing detection by mainly combining visual and text-based analysis. While attackers use various sophisticated tactics like logo obfuscation to bypass traditional systems, PhishEye uses its URL classifier since this classifier would help the model in examining the structural elements, such as, the length, subdomains, special characters, and suspicious keywords are structured in a URL [8]. This approach is effective as it identifies how deceptive

domains, like “secure-PayPal[.]com,” exist while also mimicking legitimate sites, capable of tricking consumers to a large extent. This approach also helps the model against modern phishing techniques; while also adapting to various updating threats. [7].

3.3.1.Hybrid Visual-Textual Analysis:

Though Phishpedia has high accuracy in detecting Phishing websites, it still underperforms and struggles in the identification of webpages lacking logo matches. For instance, the Text-only scams are among the most common of these and most of these scams lack appropriate logo match. Such cases then bypass detections as the system extensively relies on visual logo verifications. Thus, to address this, the BERT-based model could be integrated which assists in logo checks through the semantic structural analysis of the URLs. This solution possesses the capability of enhancing Phishing coverage by almost 10% while strengthening the detection ability of the model. However, the Trade-offs also exist as there would be an increase of 1.5 seconds delay in latency in every URL. Such minimal delay can be considered acceptable given the enhancement in detection and accuracy of the system.

Aspect	Phishpedia (R3)	PhishEye
Scope	Logo-domain consistency only	Logo + URL text analysis
Coverage	Limited to preloaded brands	Detects phishing for any brand
Explainability	Visual annotations (logo/domain)	Textual explanations (e.g., "Suspicious substring 'login'")
Adversarial Robustness	Fails on text-only/ scrambled logos	Resilient to logo removal via text analysis

Table 1: Shows Comparative Analysis of PhishEye and Phishpedia based on Detection Capabilities and Robustness

3.3.2. Faster R-CNN Selection:

As illustrated in Table 1, The model is selected over YOLO (You only look once) model as it has much higher logo detection precision in comparison to other selections. The YOLO model is one of most popular object detection tools in the world and it is known for its precision and accuracy.

3.3.3. BERT Over LSTMs/CNN’s:

In the case of BERT over CNNS/LSTMS, the transformers effectively work towards capturing long-range dependencies within the URLs much more carefully.

4. Implementations

4.1. Tools & Libraries

- **Logo Detection:** Detectron2 (Faster R-CNN) is used for object detection.
- **Logo matching:** logo matching is processed through the PyTorch-based Siamese network with triplet loss.
- **URL classifier:** Hugging Face Transformers as well as imanoop7/bert-phishing-detector model are also used within this. Target list: 181 preloaded brands (legacy).
- **Phishing Benchmark**: 30,000 URLs (30k dataset), a subset of 996 for testing.

4.2. Key Parameters

- The threshold for Logo detection is: 0.85**.**
- The similarity score for the Siamese network approach is: >0.92 for brand match.
- BERT classifier truncation would take around 512 tokens to handle long URLs.

4.3. Reproducibility

The BERT model can be publicly accessed through Hugging Face Hub and such accessibility would help developers to integrate this model easily. Additionally, the integration of Phishpedia with the Dectectron2 framework is also accessible within its repository as this supports reproducibility while also ensuring that further research within phishing detection could be conducted.

5. Performance Evaluation

5.1. Key Observations

PhishEye reduces false negatives by limiting the cases to almost 218/229 cases as illustrated in Table 2. BERT Contribution: Around 218 phishing URLs went undetected under the Phishpedia which were originally flagged by the BERT model.

System	Detected Phishing	False Negatives	Accuracy
Phishpedia	767	229	77.0%
PhishEye	985	11	98.9%

Table 2: Shows Performance Comparison of Phishpedia and PhishEye in Detecting Phishing URL

5.2. Limitations

BERT False Positives: As illustrated in Figure 3, 11 false negatives demonstrate to what extent the adversarial URLs had bypassed under the text-based checks.

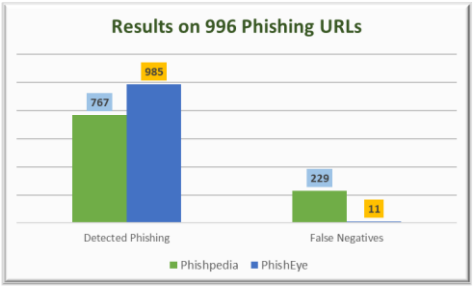

Figure-3: Shows Bar graph comparing Results of Phishpedia and PhishEye on 996 URLs

6. Results

6.1. Final Output

As per the results and analysis, it becomes apparent that PhishEye is capable to successfully achieve around 98.9% detection accuracy as per its last test set and it is able to further limit its false negatives to 11/996.

6.2. Critical Implications:

The results further showed that the broader coverage significantly enhanced detections by around 10% as this coverage detected phishing campaigns that had targeted smaller/lesser-known brands. Additionally, the hybrid defence approach was also employed as it was necessary in instances where both visual and textual analyses were required to address the limitations inherent in its logo-based detection systems.

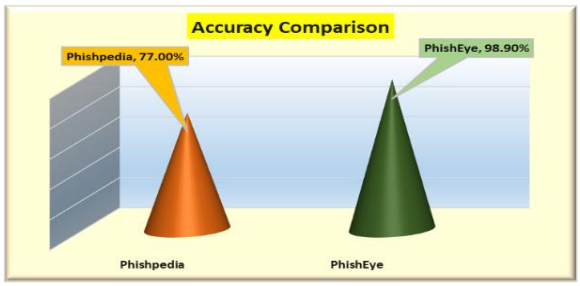

Figure-4: Shows Accuracy Comparison of Phishing Detection Tools: Phishpedia vs. PhishEye

6.3. Visualization

As illustrated in Figure 4, a substantial performance gap existed since PhishEye achieved an impressive accuracy rate of 98.90% which led it to significantly outperform Phishpedia, which had an accuracy of only 77%. Such difference showed the advantage of PhishEye in identifying phishing threats which also included deceptive or complex threats that Phishpedia bypassed. The visual comparison further shows the effectiveness of PhishEye's approach which included the integration of both textual and visual analysis while Phishpedia relied extensively on logo-based analysis for detecting phishing threats.

6.4. Improvements in PhishEye Over Phishpedia

The implementation of PhishEye is also significant as it introduces a variety of key improvements which its predecessor did not have. PhishEye further expands its brand recognition which also improves its logo detection score and ensures that it integrates adaptive learning, necessary for making brand database updates. Besides, the model had effectively implemented the Transformer-based URL analysis and with such implementation, a secondary classification and contextual understanding were adhered to in URL detections [7]. This adoption also ensures improvements in detection accuracy as such improvement can further specified for sites that don't have any recognizable logos.

In addition, the system also uses a hybrid detection framework which effectively combines visual and text-based analysis; which Phishpedia lacked [9]. The only trade-off within this model is its increased computational load which is due to Transformer analysis as it causes a slight 1.5-second delay in latency [11]. Such minor delay can be considered

acceptable given the model has enhanced Phishing detection and accuracy.

7. Critical Implementations

PhishEye plays a vital role in enhancing phishing detection as it combines visual and textual analysis while also reducing potential undetected phishing attempts. As illustrated in Figure 4, the model has an accuracy of 98% when compared to Phishpedia which is further behind at 77% [11]. Such a hybrid approach is immensely effective as it accurately distinguishes between advanced tactics like AI-generated logos and deceptive URLs. Besides, PhishEye's adaptability can further be denoted through its superiority in dealing with evolving phishing threats [7]. The model further reduces risks of data breaches and financial losses which also makes it much useful in the field of cyber security [5].

7.1. Implications for Cybersecurity and Future Work

Under its implications, PhishEye is rendered viable for use in the Security Operations Centre (SOC) since it is much more efficient in detecting phishing attacks in comparison to Phishpedia**.** Further, such a model is also important as it has immense applications for analysts who are inclined towards focusing on complex threats that they might face. A remarkable achievement also exists within this model of hybrid visual-textual analysis itself as the analysis further embarks on zero-day attacks which further strengthens the proactive defence. Due to such improvements in its accuracy, the system can further be rendered adaptable and scalable across various platforms, like email and mobile security [10].

Additionally, Future improvements would be more focused on integrating multimodal transformers and optimizing the real-time performance of the model as it would apply a variety of adversarial training for enhancing detection percentages. This vast integration would also cause immense resilience against various evolving phishing tactics throughout dynamic cybersecurity environments. Moreover, the model's modularity allows seamless integration with existing threat detection infrastructures which further help the model in reducing the deployment friction. Due to such modularity, the model becomes suitable for cybersecurity applications [9].

7. Discussion

Primarily, the PhishEye demonstrates its superiority through a significant advancement that it possesses over Phishpedia which is its hybrid two-stage framework. This framework not only integrates the visual analysis but also aids in semantic URL analysis. Its superiority is further led by its adequate use of Transformer-based models which complements visual logo-domain matching. This model not only ensures accurate detection in the instances of logos being obfuscated or absent but also enhances the overall efficiency of the tool [6]. Therefore, such integration enhances phishing detection against emerging threats which use brands or adversarial logos for carrying out phishing.

A 28.4% increase was also witnessed in the model's accuracy as due to such enhancements, a huge dramatic reduction of false negatives was also experienced within system coverage [7]. PhishEye further addresses both structure and content of URLs which renders it to be a vital component in the instance where legacy systems like Phishpedia make inaccurate predictions. Phishpedia's failure in these scenarios is of immense incumbency as it risks the credentials of millions of users; this further highlights why it's important to induce a hybrid approach in evolving cyber threat landscapes [4].

8. Related works

Recent studies explore into a variety of phishing detection methods being used for detecting phishing attacks. While some may have benefits over each other, each of them possesses a weakness which harms its accuracy in detection. Ebubekir et al. [22] introduced, ChatPhishDetector, a system uses ChatGPT for analyzing websites with an accuracy of 98.7% but such a model is much expensive in comparison to other models. The GEPAgent is another model in this field which employs GPT-4 for real-time detection by dynamically updates information using LLM models, unlike traditional methods that relays on predefined dateset [21]. Besides, the PhishLLM using predefinedis LLM models as it avoids predefined brand lists yet is extensively dependent on search engines [17]. Hao et al. [18] presented LogoMorph which is a tool that employs AI-generated logos to escape visual phishing detectors by slightly altering design aspects.

Heiding et al. [16] The V-Triad technique mainly detects phishing by comparing phishing email created by human and LLMs such as GPT-4, and shows much The study reveals that combining GPT with V-Triad produces highly convincing results, but it also identifies drawbacks such as high computing cost and false positives. Besides, Traditional ML approaches like DARTH are also used in cybersecurity fields as it combines NLP and ML for email analysis (99.97% accuracy), however, it often bypasses attachment-based attacks while a similar model, CANTINA+, uses URL/HTML features along with ML models which is also easier to bypass it due to its use of third-party services which could impact the system's flexibility and efficiency, especially when adopting to new phishing tactics [13][15].

While the majority of models rely on AI and ML factors for detection, Other methods rely much on human factors. The Phish Scale is a prominent model which detects phishing sites by rating phishing emails difficult for training yet it often makes inaccurate predictions due to individual traits being ignored [20]. Lateral phishing detection is also a similar model as it analyzes compromised accounts yet it often bypasses attachment-based attacks and because the majority of phishing attacks are attachment-based, the model becomes inefficient in the modern cyber security landscape [19].

While the LLMs review GPT-4's potential they tend to lack practical testing; even though LLMs have higher adaptability their high costs and false positives issues persist. Though Traditional ML methods balance accuracy their efficiency decreases due to the presence of zero-day threats. In comparison to these models, PhishEye emerges as a superior model among them considering all of the factors. As illustrated in Table 3, The PhishEye model achieves much higher accuracy with its only trade-off of a slight 1.5-second delay in latency which can be considered negligible as it significantly improved detection performance.

Table 3, presents a comparative summary of the various phishing detection approaches based on key evaluation criteria. As shown, current models such as PhishLLM [17], GEPAagent [21], and ChatPhishDetector [22] use AI and LLMs to detect real-time and zero-day phishing attempts. While, traditional techniques, such as multi-model frameworks [13] and CANTINA+ [15], focus on URL and HTML analysis but may be less adaptable. The table emphasizes the growing importance of artificial intelligence (AI) in modern phishing detection systems.

Paper	Technique	URL Analysis	Content Examination	Machine Learning & AI	User Behavior	Technical Indicators	Real-Time Detection	Novelty/Zero-Day Detection
1	Multi-Model Framework (ML + NLP) [13]	✓	✓	✓	✗	✓	✗	✓
2	Large Language Models (LLMs) in Cybersecurity [14]	✗	✓	✓	✗	✗	✗	✓
3	ML on URL, HTML, and Website Features [15]	✓	✓	✓	✗	✗	✓	✓
4	Phishing Creation and Detection [16]	✗	✓	✓	✗	✗	✗	✗
5	PhishLLM (LLM-based Detection) [17]	✓	✓	✓	✓	✓	✓	✓
6	Adversarial Logo Generation (LogoMorph) [18]	✗	✗	✗	✗	✓	✗	✓
7	Lateral Phishing Detection (Random Forest Classifier) [19]	✓	✓	✓	✓	✓	✓	✗
8	Phishing Detection Difficulty Rating [20]	✓	✗	✓	✗	✗	✗	✗
9	AI-based Detection (GEPAgent) [21]	✓	✓	✓	✓	✓	✓	✓
10	AI-based Detection (ChatGPT) [22]	✓	✓	✓	✗	✗	✓	✓

Table 3: Shows comparative Analysis of Phishing Detection Methods in related works.

9. Threats to Validity

Despite its efficiency, the PhishEye model faces a variety of threats which harm its validity. Primarily, the dataset bias is one of the most common biases which exist due to benchmark URLs as it does not showcase distributions adequately. Secondly, overfitting in the model is also a major threat as it only tends to occur in the case of over-reliance on training data patterns [4]. Besides, the existence of false positives generated by BERT-based classifiers can also be considered a threat to validity since the benign URLs are wrongly flagged due to complexities in phishing patterns. Additionally, due to second stage Transformers, the latency increases slightly which potentially threatens the probablity of scalability [3].

10. Conclusion

Phishing is considered to be among the most significant cybersecurity threats nowadays as the attacker possesses the ability to create fake websites that mimic real services with the one main purpose of stealing personal data. Even in these phishing attacks, the dynamic nature of attacks renders various machine learning analyses to be evaded easily. Besides, The PhishEye model can be effectively used in this process through mainly targeting deceptive URLs and fake branding tactics which most attacks use to mimic the authenticity of the actual brand. Thus, it can be denoted that the use of the PhishEye model improves the protection against such sophisticated phishing attempts when compared with other models. This model further proceeds with the textual and structural-based URL analysis as such analysis would improve

the classification process in the instances where the logos are unavailable. Such minimal delay can be considered acceptable given the enhancement in detection and accuracy of the system. These models as further offered within Hugging Face's Transformers library.

References

[1]. A. A. Orunsolu, A. S. Sodiya, and A. T. Akinwale, "A predictive model for phishing detection," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 2, pp. 232-247, 2022. https://www.sciencedirect.com/science/article/pii/S1319157819304902

[2]. A. A. Zuraiq and M. Alkasassbeh, "Phishing detection approaches," in *Proc. 2nd Int. Conf. New Trends in Computing Sciences (ICTCS)*, Oct. 2019, pp. 1-6. https://www.researchgate.net/publication/337787302_Review_Phishing_Detection_Approaches

[3]. B. Srikanth, "AI-Powered Phishing Detection: Protecting Enterprises from Advanced Social Engineering Attacks," 2022. https://philpapers.org/rec/SRIAPD

[4]. D. Rathee and S. Mann, "Detection of E-mail phishing attacks–using machine learning and deep learning," *Int. J. Computer Applications*, vol. 183, no. 1, p. 7, 2022. https://www.academia.edu/download/79535611/rathee_2022_ijca_921868.pdf

[5]. G. G. Geng, X. D. Lee, and Y. M. Zhang, "Combating phishing attacks via brand identity and authorization features," *Security and Communication Networks*, vol. 8, no. 6, pp. 888-898, 2015. https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1045

[6]. G. Varshney, M. Misra, and P. K. Atrey, "A survey and classification of web phishing detection schemes," *Security and Communication Networks*, vol. 9, no. 18, pp. 6266-6284, 2016. https://onlinelibrary.wiley.com/doi/abs/10.1002/sec.1674

[7]. K. W. Kwaku, "AI-Powered Phishing Detection Systems: Challenges and Innovations," *Advances in Computer Sciences*, vol. 5, no. 1, 2022. https://acadexpinnara.com/index.php/acs/article/view/278

[8]. M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2091-2121, 2013. https://ieeexplore.ieee.org/abstract/document/6497928/

[9]. O. A. Lamina, W. A. Ayuba, O. E. Adebiyi, G. E. Michael, O. O. D. Samuel, and K. O. Samuel, "AI-Powered Phishing Detection and Prevention," *Path of Science*, vol. 10, no. 12, pp. 4001-4010, 2024. https://pathofscience.org/index.php/ps/article/view/3406

[10]. O. K. Sahingoz, E. Buber, O. Demir, and B. Diri, "Machine learning based phishing detection from URLs," *Expert Systems with Applications*, vol. 117, pp. 345-357, 2019. https://www.sciencedirect.com/science/article/pii/S0957417418306067

[11]. V. Shahrivari, M. M. Darabi, and M. Izadi, "Phishing detection using machine learning techniques," *arXiv preprint arXiv:2009.11116*, 2020. https://arxiv.org/abs/2009.11116

[12]. Y. Lin, R. Liu, D. M. Divakaran, J. Y. Ng, Q. Z. Chan, Y. Lu, Y. Si, and F. Zhang, "Phishpedia: A hybrid deep learning based approach to visually identify phishing webpages," in *Proc. 30th USENIX Security Symp. (USENIX Security 21)*, Aug. 2021, pp. 379–396 https://www.usenix.org/conference/usenixsecurity21/presentation/lin

[13]. Mittal, Apurv, Daniel Engels, Harsha Kommanapalli, Ravi Sivaraman, and Taifur Chowdhury. "Phishing Detection Using Natural Language Processing and Machine Learning." *SMU Data Science Review*, vol. 5, no. 1, 2022. Retrieved from https://scholar.smu.edu/cgi/viewcontent.cgi?article=1215&context=datasciencereview.

[14]. Divakaran, D. M., & Peddinti, S. T. (2024). *LLMs for Cyber Security: New Opportunities*. arXiv. https://arxiv.org/pdf/2404.11338

[15]. Xiang, Guang, et al. "CANTINA+: A Feature-rich Machine Learning Framework for Detecting Phishing Websites." *Carnegie Mellon University Digital Archive*, 2019. Retrieved from https://www.ml.cmu.edu/research/dap-papers/dap-guang-xiang.pdf.

[16]. Heiding, M. "Devising and Detecting Phishing: Large Language Models vs. Smaller Human Models." *Black Hat USA*, 2023. Retrieved from https://i.blackhat.com/BH-US-23/Presentations/US-23-Heiding-Devicing-and-Detecting-Phishing-wp.pdf.

[17]. Liu, Ruofan, et al. "Less Defined Knowledge and More True Alarms: Reference-based Phishing Detection without a Pre-defined Reference List." *USENIX Security Symposium*, 2024. Retrieved from https://www.usenix.org/conference/usenixsecurity24/presentation/liu-ruofan.

[18]. Hao, Qingying, et al. "It Doesn't Look Like Anything to Me: Using Diffusion Model to Subvert Visual Phishing Detectors." *USENIX Security Symposium*, 2024. Retrieved from

https://www.usenix.org/system/files/usenixsecurity24-hao-qingying.pdf.

[19]. Ho, Grant, Asaf Cidon, Lior Gavish, Marco Schweighauser, Vern Paxson, Stefan Savage, Geoffrey M. Voelker, and David Wagner. "Detecting and Characterizing Lateral Phishing at Scale." *USENIX Security Symposium*, 2019. Distinguished Paper Award Winner. Retrieved from https://www.usenix.org/conference/usenixsecurity19/presentation/ho.

[20]. Steves, Michelle P., et al. "A Phish Scale: Rating Human Phishing Message Detection Difficulty." *NDSS Symposium*, 2019. Retrieved from https://www.ndss-symposium.org/wp-content/uploads/2019/02/usec2019_02-4_Steves_paper.pdf.

[21]. Ebubekir, B., et al. "Automated Phishing Detection Using URLs and Webpages." *arXiv*, 2024. Retrieved from https://arxiv.org/pdf/.

[22]. Ebubekir, B., et al. "Detecting Phishing Sites Using ChatGPT." *arXiv*, 2023. Retrieved from https://arxiv.org/pdf/2306.05816.