Literature Review:

Data	Authors & Year	Main Idea	Findings & Results	Dataset used	Limitations
URL	Al-Ahmadi et al, 2022	This paper proposes a model (PDGAN) to detect phishing websites using Generative Adversarial Networks (GANs). They used LSTM to generate synthetic phishing URLs, and then employed CNN for classification.	The PDGAN demonstrated superior performance compared to other methods, boasting a low False Positive (FP) Rate of 2.1% and a F1 score of 97.64%.	The authors obtain an extensive dataset comprising almost 2 million URLs from DomCop and PhishTank websites.	The generator should produce synthetic phishing URLs that are similar to the real URLs and is based on the assumption that the discriminator can distinguish between legitimate and phishing URLs.
	Peng et al, 2019	In this paper, the authors introduce an approach merging an Attention mechanism with a CNN and LSTM to identify malicious URLs. They use WHOIS check method to extract and filter features, and subsequently input them to the constructed CNN convolution layer to extract local features.	The key highlight of this paper is the introduction of attention mechanism in the expression of the malicious URL feature allows for the highlighting of key features to detect malicious URLs.	The dataset is collected from PhishTank and contains 16,055 malicious URLs.	The paper does not provide any report or analysis on the dataset or its collection and is missing the benign instances.

	Bahnsen et al, 2017	This study introduces a novel method utilizing RNN to classify phishing URLs. This approach eliminates the need for manual feature extraction by directly acquiring a representation from the sequence of characters within the URL. More precisely, the model employs LSTM units, treating the URL as a character sequence, where each character is transformed into a 128-dimensional embedding as input.	The LSTM model achieved an accuracy rate of 98.7%, outperforming the RF model, which achieved an accuracy rate of 93.5%. On average, the LSTM network has an F1 score 5% higher than the feature engineer model with RF. This paper also highlights that phishing URLs tend to be longer than benign URLs. It also highlights the benefits of utilizing representations of URL character sequences to classify websites.	For their experiments, a dataset of real and phishing URLs was created consisting 2 million instances. The dataset had a balanced distribution. The legitimate URLs came from Common Crawl whereas the phishing URLs came from Phishtank, a website used as phishing URL deposit.	The core concept of the paper involves translating each input character into a 128-dimensional embedding, which is then inputted into LSTM units. This hypothesis is derived from the assumption that characters close to each other in a URL are likely to be correlated.
	Xiang et al, 2011	The paper proposes a novel anti-phishing solution named CANTINA+ that aims to address the weaknesses of both blacklists and feature-based methods in a unified framework. It contains three major modules. First leverages the high similarity among phishing webpages. The second detects the presence of login forms that request sensitive information. The third and core module introduces 15 highly expressive features with ML algorithms to classify webpages.	This study presents eight innovative features that improve the detection of phishing attacks by utilizing diverse resources. It tackles the challenge of high FPs by using a layered structure with login form filtering. CANTINA+ attained high TP rates of 92.54% and 93.47%, accompanied by low FP rates of 0.407% and 0.608% with and without login form filtering, respectively. However, the authors note that the filtering step notably decreases the TP rate.	The webpage collection consists of phishing cases from PhishTank, and legitimate webpages from five sources.	The disadvantage of this solution is the use of a search engine to find out whether the address certainly matches the desired page causing additional network load. Moreover, the attacker could promote the phishing website in the search engine to make it seem legitimate.

Image	Ouyang and Zhang, 2021	The authors introduce an innovative approach for detecting phishing web pages, utilizing a graph neural network framework. They utilize the HTML's inherent DOM tree structure, representing it as a graph to extract local features from each node through RNN and distributed representations. This solution merges the strengths of RNN for extracting local features with GNN's ability to capture broader semantic context.	The authors introduce their approach as the first study into modeling HTML codes as graphs and classifying web pages using GNNs. Their proposed solution surpasses current anti-phishing methods, achieving an outstanding accuracy of 95.5%. The model demonstrates its optimal performance when utilizing information extracted from the HTML DOM tree graphs.	Phishing web pages were obtained from PhishTank and OpenPhish. In order to obtain the HTML content of these pages, they developed a custom crawler. For benign web pages, they utlize the TrancoTop1M which contains top 1 million domain names by traffic. The authors use it this as a starting point and crawl web pages by following the links on them. Their final dataset contisted of 26,578 phishing and 121,983 benign instances.	The experiments conducted by the authors primarily aim at generating the DOM structure of HTMLs which may not be applicable for all types of web pages. The computational resources required for the proposed method are not discussed, limiting its applicability to resource constrained environments.
	Wei et al, 2020	In this paper, the authors present a method to detect malicious URL addresses with almost 100% accuracy using CNN. Their method includes encoding the URLs using one-hot character-level representation of URLs to create input images for the neural network. the proposed method aims to provide nearly 100% accurate security, zero-day defense, and fast detection, making it suitable for real-time URL checking and mobile device usage.	Key finindings include the use of an embedding layer to improve the accuracy of the proposed method. Employing CNNs with character-level URL encoding resulted in nearly perfect accuracy, reaching close to 100% in identifying phishing URLs, surpassing other existing methods relying on engineered features or content analysis.	The experiments are based on a publicly available PhishTank phishing sites database and benign URLs from CommonCrawl. The authors collected 10,604 random unique URLs of each category.	The authors made decisions, such as imposing a character limit of 256, without offering any supoorting explanations. Furthermore, the authors did not consider the presence of special symbols such as '@' or '%' in URLs, which could indicate the use of encoding and obfuscation techniques.

	Jain & Gupta, 2017 *[22]*	This article's primary aim is to offer an overview of visual similarity techniques used in identifying phishing websites. It utilizes classifiers like Earth Mover's Distance (EMD) based image classifier. Image properties serve as a component of the feature set to compare against legitimate websites. Additionally, the phishing detection system suggested here employs a signature generated from the webpage's text, images, and its overall visual presentation.	The authors integrate hybrid features, including text properties extracted from textual content and image properties such as color histograms for creating signatures to compare webpages. They classify the visual similarity based approaches into HTML document object model (DOM) tree, visual features, Cascading Style Sheet (CSS) similarity, pixel based, visual perception, and hybrid approaches.	The paper does not conduct any experiments explicitly and hence do not use any dataset. Although, the authors present a comprehensive analysis of phishing attacks and some of the recent visual similarity based approaches for phishing detection.	The authors do not present any quantifiable results or experiment that can help evaluate the proposed methods. A comparison with the previous works mentioned could have provided more insight into the various methods and feature sets used.
Code	Canali et al, 2011	Prophiler was developed as a filter for large scale detection of malicious webpages. It inspects two main sources of information for features, the content of the page (HTML and JavaScript code) as well as the associated URL (lexical and host-based characteristics). The authors developed Prophiler as a filter that can reduce the number of web pages that need to be analyzed dynamically to identify malicious web pages by tools such as Wepawet.	The outcomes obtained by Prophiler align with the authors' objectives. They conducted a large-scale assessment by deploying Prophiler across the dataset used. Prophiler identified 14.3% of these pages as malicious, leading to an 85.7% reduction in the workload for the back-end analyzer. Additionally, their solution demonstrated superior performance compared to previous systems, showcasing lower rates of false negatives and false positives.	This experimentation used two datasets: an evaluation dataset and a validation dataset. The evaluation dataset consisted of 18,939,908 pages, all of which were unlabeled. The validation dataset contained 153,115 pages, with 139,321 benign and 13,794 malicious.	The study focuses on the design and implementation of a fast filter for detecting malicious web pages, but does not provide a comprehensive analysis of the effectiveness of the filter. The study does not address the detection of zero-day exploits or new types of malicious code that may not be captured by the filter.

	Rao, Umarekar and Pais, 2021	This paper introduces a new approach for identifying phishing websites, employing word embeddings derived from both plain and domain specific text extracted from HTML source code. They utilize word embedding techniques like Word2Vec and Glove and evaluate their model using an ensemble by combining different word embedding algorithms with RF and LR classifiers for its purposes. Additionally, they utilize various traditional classifiers to compare the effectiveness of their approach.	The word embeddings are generated using both, the plain texts and domain specific texts. The results obtained reinforce the usage of domain specific texts as all classifiers perform better when using it rather than plain text. It is also observed that the proposed multimodal model outperformed existing works with a significant accuracy of 99.34% and Mathews Correlation Coefficient (MCC) of 98.68%.	The approach involves utilizing a dataset comprising HTML source code extracted from various websites. The dataset is composed of 5076 instances classified as benign and 5438 instances classified as phishing. The source codes undergo parsing, and word embedding algorithms are applied to process the text. The division between training and testing sets is performed with an 80% - 20% split.	The model relies on textual data, both general and domain-specific, and encounters challenges when presented with images instead of text. However, details regarding the dataset's availability on the internet and the specific technique employed for its collection are not provided.
	Zhang et al, 2021	The authors of this paper introduce CrawlPhish, a framework for large scale detection and categorization of client-side cloaking techniques used by known phishing websites. They take into account both the visual and code structure components of phishing websites. The code structure includes features such as web API calls, web event listeners, and ASTs. The visual similarity between force-executed screenshots and an	A major contribution of this paper was a proposed taxonomy of eight types of evasion techniques in three categories (User Interaction,Bot Behavior, Fingerprinting). The CrawlPhish framework exhibited low false-positive and false-negative rates of 1.45% and 1.75% respectively. They also identified 1,128 distinct implementations of cloaking techniques.	A dataset of 112,005 phishing websites was collected and analyzed over a 14-month period. Out of these phishing websites, 35,067 (31.3%) were found to use client-side cloaking techniques. The use of client-side cloaking by attackers increased from 23.32% in 2018 to 33.70% in 2019.	Some limitations of this study is that it does not provide insights into the prevalence and impact of advanced client-side cloaking techniques. The evaluation of the effectiveness of the cloaking techniques is also limited to browser-based phishing detection and does not consider other anti-phishing defenses.

		unmodified WebKitGTK+ screenshot is also examined by CrawlPhish.			
	Li et al, 2019	This paper introduces a stacking model designed for the classification of phishing webpages, incorporating both URL and HTML features. The authors extract 12 features from the HTML source code, specifically advocating for the use of the Word2Vec model to generate string embeddings from HTML. These embeddings and features are then fed into a stacking model, employing Gradient Boosting Decision Tree (GBDT), XGBoost, and LightGBM for the ensemble model.	The presented approach was compared against various baselines, including CANTINA, and demonstrated the lowest false alarm rate at 3.7%. To showcase the efficacy of their stacking model, a comparison was conducted against individual machine learning models using the results from the 50K-PD dataset. The proposed model surpasses all individual models, achieving an accuracy of 97.30%. Their analysis highlights the effectiveness of employing HTML string embeddings generated by Word2Vec, requiring no domain knowledge of phishing.	The authors employed three datasets to assess their methodology. Initially, they utilized the 2K Phishing Detection Dataset (2K-PD), a well-balanced collection consisting of 1000 legitimate websites and 1000 phishing websites, each accompanied by their respective HTML code. Legitimate websites were sourced from Alexa rankings, while phishing instances were gathered from PhishTank. Additionally, the authors incorporated the 50K Phishing Detection Dataset (50K-PD) and the 50K Image Phishing Detection Dataset (50K-IPD) into their evaluation.	Phishing instances are sourced exclusively from PhishTank for a brief period, specifically over a span of 2 days. This approach may result in data homogeneity and does not encompass the diverse array of sources for phishing attacks.

The field of phishing detection has been an area of research for a long time. Over the years, it has become a never-ending game of cat and mouse between the developers trying to identify and stop phishing attempts and the attackers who are constantly evolving their approaches. The blacklist/whitelist-based strategy is a popular technique in the area. It uses a list of known phishing websites, including URLs, IP addresses, and other information. However, because these lists had to be updated on a frequent basis, they were unable to handle the massive amount of webpages on the internet. [1][2]

Phishing detection has advanced significantly as a result of the growing application of machine learning and deep learning in cybersecurity. Several studies [3-22] have been conducted using URLs, website images, website source codes, and phishing kits to extract statistical, lexical, and domain-specific features, image representations, and word embeddings, which are then used to train machine learning and deep learning algorithms. These algorithms have demonstrated remarkable efficacy in identifying phishing attempts using various datasets. The table provided lists some studies conducted in this area. These are broadly categorized according to the resources they use for feature extraction, representation and embedding generation, or model training purposes. For the purpose of this study, we focus on the use of URLs, images, and source code files in the existing literature and analyze the results of these approaches.

URL-based methods: Researchers have utilized the findings to support the notion that phishing website URLs exhibit intricate and distinctive characteristics. These features can be harnessed to derive relevant attributes for the classification of phishing websites using machine learning models. Extensive work has been conducted by Mourtaji et al [3], Sahingoz et al [4], Ucar et al [5], and Afzal et al [6]. Their research involves extracting statistics-based heuristics, including URL length, the number of special symbols (‘.’, ‘@’, ‘-’) in the URL, and the presence of redirection links in the website source. Additionally, studies by Cheng et al [7], Verma et al [8], and Peng et al [9] primarily focus on generating features through expert knowledge and lexical analysis of the URL. These approaches also incorporate third-party features, such as Alexa page rankings and domain information obtained from WHOIS records.

CANTINA+ [10], an advanced anti-phishing solution, presents itself as a 'Feature-rich Machine Learning Framework' because of its comprehensive approach to detecting phishing websites through its elaborate features. This studies yielded encouraging results after a thorough review of the various features. Canali et al [11]. introduced Prophiler in their study, which employs a total of 33 features derived from the analysis of URL and host information and may be broadly classified as syntactical, DNS-based, whois-based, and geoIP-based. Their findings also highlight the relvance of such features for training machine learning systems. Bahnsen et al [12] also extract relevant features from the URLs to train an ML model. However, they introduce an innovative approach to represent the characters in a URL. This involves translating them into 128-dimensional embeddings and subsequently inputting this information into an LSTM network. Al-Ahmadi et al [13] proposed a deep learning based method named *PDGAN* that depends on the website’s URL to achieve reliable performance. It leverages the generative adversarial network (GAN) that comprises of two components, a generator made of LSTM network to create synthetic phishing URLs and a CNN as a discriminator to decide whether URLs are phishing or legitimate.

Image-based methods: With the advent of internet, attackers are able to replicate webpages from legitimate websites with a high degree of visual similarity. This increases the vulnerability of users to falling victim to phishing attacks. In response to such techniques, researchers have conducted extensive investigations into the impact of utilizing images for the identification of phishing websites. Haruta et al [14] focused on elements that determine the visual content of websites, such as images, their positions, colors, fonts, and more. Hara et al [15] employed ImgSeek to assess the similarity between image pairs from authentic webpages and their corresponding phishing counterparts. Their approach achieved a detection rate exceeding 80% for phishing images.

In more recent developments, Wei et al [16] and Ouyang et al [17] have employed advanced deep learning techniques in their research endeavors. Wei et al transformed URLs into images through character-level one-hot encoding and the subsequent generation of representations. On the other hand, Ouyang et al utilized graphs derived from the Document Object Model (DOM) tree of HTML source codes as a basis for their analysis.

Code-based methods: Roopak and Thomas [18] introduced a novel method that relied on HTML source code for creating a mechanism to detect phishing pages. In their proposed solution, they initially conduct a comparison of webpages through attribute matching of HTML tags. Subsequently, they assess the textual content of the pages to determine their cosine similarity. However, their approach may not work under code obfuscation. Li et al [19] employ HTML source code for extracting pertinent features in their proposed stacking model. Their research emphasizes the combined impact of feature extraction from both URLs and HTML. Additionally, they highlight the significance of HTML string embeddings generated through Word2Vec in enhancing the detection of phishing websites.

Zhang et al [20] introduced CrawlPhish, a framework designed for the automated detection and categorization of client-side cloaking techniques employed by phishing websites. Their approach involves leveraging JavaScript files to extract code structure features, along with visual features obtained after exploring every possible execution path through forced execution. However, due to the challenges associated with obtaining JavaScript source code files, researchers resort to utilizing more readily accessible HTML files. More recently, Rao et al [21] introduced a novel method for detecting phishing sites, employing word embeddings derived from plain text and domain-specific text extracted from the source code. The utilized word embedding algorithms were classified into frequency-based models like TF-IDF and prediction-based models such as Word2Vec and GloVe. While their results showcase a promising potential for this technique, it's essential to note that the evaluation was conducted on a relatively small dataset.

References:

[1] Cao, Y., Han, W., & Le, Y. (2008). “Anti-phishing based on automated individual white-list”. In Proceedings of the 4th ACM workshop ondigital identity.

[2] A. Oest, Y. Safaei, A. Doupé, G. -J. Ahn, B. Wardman and K. Tyers, "PhishFarm: A Scalable Framework for Measuring the Effectiveness of Evasion Techniques against Browser Phishing Blacklists," 2019 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2019, pp. 1344-1361, doi: 10.1109/SP.2019.00049.

[3] Youness Mourtaji, Mohammed Bouhorma, Daniyal Alghazzawi, Ghadah Aldabbagh, Abdullah Alghamdi, "Hybrid Rule-Based Solution for Phishing URL Detection Using Convolutional Neural Network", Wireless Communications and Mobile Computing, vol. 2021, Article ID 8241104, 24 pages, 2021. https://doi.org/10.1155/2021/8241104.

[4] Ozgur Koray Sahingoz, Ebubekir Buber, Onder Demir, Banu Diri, “Machine learning based phishing detection from URLs”, Expert Systems with Applications, Volume 117, 2019, Pages 345-357, ISSN 0957-4174. https://doi.org/10.1016/j.eswa.2018.09.029.

[5] E. Ucar, M. Ucar, and M. O. Incetas ̧, “A deep learning approach for de-tection of malicious urls,” in 6th International Management Information Systems Conference, 2019, pp. 12–20.

[6] Afzal, S., Asim, M., Javed, A. R., Beg, M. O., & Baker, T. (2021). “URLdeepDetect: A Deep Learning Approach for Detecting Malicious URLs Using Semantic Vector Models”. Journal of Network and Systems Management, 29(3). doi:10.1007/s10922-021-09587-8

[7] Yanan Cheng, Tingting Chai, Zhaoxin Zhang, Keyu Lu, Yuejin Du, “Detecting Malicious Domain Names with Abnormal WHOIS Records Using Feature-Based Rules”. The Computer Journal, Volume 65, Issue 9, September 2022, Pages 2262–2275. https://doi.org/10.1093/comjnl/bxab062

[8] R. Verma and K. Dyer, “On the Character of Phishing URLs: Accurate and Robust Statistical Learning Classifiers,” in ACM Conference on Data and Application Security and Privacy, 2015, pp. 111–121.

[9] Y. Peng, S. Tian, L. Yu, Y. Lv, R. Wang, "MALICIOUS URL RECOGNITION AND DETECTION USING ATTENTION-BASED CNN-LSTM," KSII Transactions on Internet and Information Systems, vol. 13, no. 11, pp. 5580-5593, 2019. DOI: 10.3837/tiis.2019.11.017.

[10] Guang Xiang, Jason Hong, Carolyn P. Rose, and Lorrie Cranor. 2011. CANTINA+: A Feature-Rich Machine Learning Framework for Detecting Phishing Web Sites. ACM Trans. Inf. Syst. Secur. 14, 2, Article 21 (September 2011), 28 pages. https://doi.org/10.1145/2019599.2019606

[11] Davide Canali, Marco Cova, Giovanni Vigna, and Christopher Kruegel. 2011. Prophiler: a fast filter for the large-scale detection of malicious web pages. In Proceedings of the 20th international conference

on World wide web (WWW '11). Association for Computing Machinery, New York, NY, USA, 197–206. https://doi.org/10.1145/1963405.1963436

[12] A. C. Bahnsen, E. C. Bohorquez, S. Villegas, J. Vargas and F. A. González, "Classifying phishing URLs using recurrent neural networks," 2017 APWG Symposium on Electronic Crime Research (eCrime), Scottsdale, AZ, USA, 2017, pp. 1-8, doi: 10.1109/ECRIME.2017.7945048.

[13] S. Al-Ahmadi, A. Alotaibi and O. Alsaleh, "PDGAN: Phishing Detection With Generative Adversarial Networks," in IEEE Access, vol. 10, pp. 42459-42468, 2022, doi: 10.1109/ACCESS.2022.3168235.

[14] S. Haruta, H. Asahina and I. Sasase, "Visual Similarity-Based Phishing Detection Scheme Using Image and CSS with Target Website Finder," GLOBECOM 2017 - 2017 IEEE Global Communications Conference, Singapore, 2017, pp. 1-6, doi: 10.1109/GLOCOM.2017.8254506.

[15] M. Hara, A. Yamada and Y. Miyake, "Visual similarity-based phishing detection without victim site information," 2009 IEEE Symposium on Computational Intelligence in Cyber Security, Nashville, TN, USA, 2009, pp. 30-36, doi: 10.1109/CICYBS.2009.4925087.

[16] Wei Wei, Qiao Ke, Jakub Nowak, Marcin Korytkowski, Rafał Scherer, Marcin Woźniak, Accurate and fast URL phishing detector: A convolutional neural network approach, Computer Networks, Volume 178, 2020, 107275, ISSN 1389-1286. https://doi.org/10.1016/j.comnet.2020.107275.

[17] L. Ouyang and Y. Zhang, "Phishing Web Page Detection with HTML-Level Graph Neural Network," 2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Shenyang, China, 2021, pp. 952-958, doi: 10.1109/TrustCom53373.2021.00133.

[18] S. Roopak and T. Thomas, "A Novel Phishing Page Detection Mechanism Using HTML Source Code Comparison and Cosine Similarity," 2014 Fourth International Conference on Advances in Computing and Communications, Cochin, India, 2014, pp. 167-170, doi: 10.1109/ICACC.2014.47.

[19] Yukun Li, Zhenguo Yang, Xu Chen, Huaping Yuan, Wenyin Liu, “A stacking model using URL and HTML features for phishing webpage detection”, Future Generation Computer Systems,nVolume 94, 2019, Pages 27-39, ISSN 0167-739X. https://doi.org/10.1016/j.future.2018.11.004.

[20] P. Zhang et al., "CrawlPhish: Large-scale Analysis of Client-side Cloaking Techniques in Phishing," 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021, pp. 1109-1124, doi: 10.1109/SP40001.2021.00021.

[21] Rao, R.S., Umarekar, A. & Pais, A.R. Application of word embedding and machine learning in detecting phishing websites. Telecommun Syst 79, 33–45 (2022). https://doi.org/10.1007/s11235-021-00850-6

[22] Ankit Kumar Jain, B. B. Gupta, "Phishing Detection: Analysis of Visual Similarity Based Approaches", Security and Communication Networks, vol. 2017, Article ID 5421046, 20 pages, 2017. https://doi.org/10.1155/2017/5421046