A COMPREHENSIVE DATASET FOR WEBPAGE CLASSIFICATION

M. Al-Maamari, M. Istaiti, S. Zerhoudi,
M. Dinzinger, M. Granitzer, J. Mitrovic
University of Passau, 94032 Passau, Germany

Abstract

While webpage classification may not be a fundamental requirement for basic web crawling, it proves useful in enhancing the prioritization of crawled webpages. In this regard, our study presents a dataset of 116,000 URLs, complete with their content, specifically curated for webpage classification tasks. The primary goal of this research is to establish this comprehensive dataset with two levels of labels for URLs. Firstly, a broad level categorization dividing URLs into Malicious, Benign, or Adult, and secondly, a more nuanced labeling which includes 20 subclasses, providing a more granular view of the webpage content.

The secondary objective is to leverage this dataset for testing and comparing the performance of various machine learning models, specifically Stochastic Gradient Descent (SGD) and Support Vector Classifier (SVC), in the task of webpage classification. This involves investigating the effectiveness of different input types (URLs only, raw HTML content, and parsed HTML content) and various tokenization methods (character-level, word-level, Byte Pair Encoding(BPE) [1]) on model performance.

A total of 36 experiments were conducted, yielding several important findings. Using only the URL as input consistently resulted in the highest F1 score 0.94. Character-level tokenization consistently outperformed other tokenization techniques. There was a negligible difference in the accuracy of webpage classification between SGD and SVC models.

This research's findings demonstrate the viability of URL-based classification systems in web crawlers and shed light on optimal techniques for feature representation. The comprehensive dataset and results presented in this paper make valuable contributions to the advancement of web crawling applications, especially those requiring effective content prioritization and filtering.

INTRODUCTION

With the exponential growth of the World Wide Web, the number of web pages being created each day has reached unprecedented levels. According to recent statistics up until August 2021, the total number of websites had surpassed 1.88 billion[1], and this number continues to rise rapidly. In such a vast digital landscape, it becomes increasingly challenging for search engines, web crawlers, and other automated systems to efficiently navigate and extract relevant information, especially within the confines of a closed search ecosystem dominated by a few gatekeepers [2]. From traditional methods that analyze keywords, HTML structures, and link patterns, to advanced techniques leveraging machine learning and natural language processing, researchers and developers are continuously refining classification algorithms.

To address this challenge, the classification of web pages using URLs and HTML has emerged as a promising approach. By organizing and categorizing web pages, this technique enables improved crawling efficiency, enhanced search results, and more targeted content indexing. In this article, we delve into the significance of web page classification and its potential benefits for crawling operations.

In our research, we present a comprehensive dataset comprising two distinct groups for web page classification. The dataset includes the URL of the web page, along with the corresponding HTML content and the content extracted from the HTML without any markup. The first group consists of three main categories: Malicious, Adults, and Benign. This grouping allows for the identification and categorization of web pages with potentially harmful or explicit content, as well as those that are considered safe and harmless.

Moreover, within the second group of our dataset, we have further subdivided the benign category into various subgroups. These subgroups include topics such as sports, news, kids, and more. This finer-grained classification enables a more precise targeting and categorization of web pages based on their specific content and themes.

This research has made a contribution by creating a comprehensive dataset that combines URLs with their corresponding HTML content. This dataset serves as a valuable resource for training machine learning models to classify web pages. By utilizing this dataset, we conducted a comparative analysis of classification approaches using both URL characteristics and the content of web pages. This investigation revealed the efficacy of using URLs over webpages' content as input to the classification models, showcasing the potential for more accurate and targeted web page categorization. The findings of this study highlight the difference between using URL and content in web page classification and provide valuable insights for improving crawling efficiency and enhancing the overall performance of automated systems in navigating the vast digital landscape.

RELATED WORK

Webpage classification using Uniform Resource Locators (URLs) represents a critical area of study in the realm of information retrieval, web mining, and cybersecurity [3, 4]. Researchers have proposed diverse strategies for this task, emphasizing distinct aspects such as the linguistic features within URLs, efficiency considerations, or integration with HTML content. As the prime interface between users and

[1] https://www.statista.com/cart/19058/number-of-websites-online/

web resources, URLs contain significant information about the content of webpages, making them a valuable feature for webpage classification tasks. Understanding the various ways in which URLs have been leveraged for classification, as well as the different methodologies employed, provides vital context and inspiration for further exploration in this area.

One innovative approach has been introduced by Abdallah and de La Iglesia [5]. In their paper, they present the argument that URLs, albeit brief, offer a wealth of information for classification tasks, including potentially domain-specific terminology and abbreviations. They identified the inefficiencies in the brute-force approach, which extracts all possible substrings (allgrams) as the classifier's feature set, due to its inability to scale well for large datasets. In response, they proposed an n-gram language model for webpage classification, introducing an efficient method that not only offers competitive accuracy but also ensures scalability. Their technique, borrowing the concept of language models from the fields of information retrieval and automatic speech recognition, has shown promising results on multiple datasets with different classification objectives, where they achieved 0.82 F1 score in the DMOZ dataset, illustrating its potential utility in a wide range of URL classification scenarios.

Min-Yen Kan and Hoang Oanh Nguyen Thi [4], have advanced the field of webpage classification by developing a unique method that accelerates the classification process, only using URLs as the source of input. Their methodology involves segmenting the URL into meaningful components and extracting salient patterns to be used in supervised maximum entropy modeling. They demonstrated the effectiveness of this approach by showcasing its performance against a full-text standardized dataset (WebKB). The results were promising with F1 score of 0.62, indicating that integrating URL-based features alongside the content of the webpages can indeed match or even surpass previous methods. This work highlights the potential of using URLs as a robust and efficient feature for webpage classification.

The work by Ali Aljofey [6] experiences into a critical domain of security by focusing on phishing website detection. This research uses both URLs and HTML content to extract features, categorizing them into four groups. Some of the features presented are newly proposed, reflecting the continual innovation in this field. Aljofey's approach also demonstrates the utility of machine learning techniques in webpage classification tasks, with their results on a costume dataset they used, showing a high F1 score of 0.96, particularly when the XGBoost classifier was applied on a combination of all the features.

Lastly, the work by Hung Le and colleagues [7] provides a remarkable contribution to detecting malicious URLs through deep learning. They propose URLNet, a framework designed to overcome the shortcomings of traditional methods that primarily rely on blacklists. Their method uses convolutional neural networks to capture semantic information and sequential patterns in URLs. Their results reveal the impressive performance of URLNet in terms of significant improvements over baseline methods across various metrics, where they tested their framework on a large dataset collected from VirusTotal, and got an accuracy of 0.99, making their work an influential reference in the study of webpage classification using URLs.

Despite the considerable progress made in classifying webpages based on their URLs and HTML content, as aforementioned, a critical challenge remains in the form of the availability of adequately annotated datasets that can enable researchers to train and evaluate novel classification models. Recognizing this gap, our work introduces a comprehensive and openly available dataset, with 116,000 URLs, complete with raw HTML and parsed content. We have provided two levels of labels, firstly, a broad level categorizing URLs as Malicious, Benign, or Adult, and secondly, a more nuanced labeling which includes categories like 'Spam', 'Malware', 'Society', and 'Arts'. This extensive dataset with multiple levels of labeling not only addresses a significant gap in the field but also enables the development and evaluation of more sophisticated webpage classification models. Moreover, by comparing the performance of different machine learning models, we provide additional insights into the potential of different data representations and tokenization methods for webpage classification.

METHODOLOGY

In this section, we outline the methodology employed in our study for building a dataset of URLs of webpages and their HTML content, then use this dataset to classify web pages based on their URL, raw HTML content, and parsed content. The goal is to identify various categories of web pages, including benign, adult, and malicious. We present the process of data collection, where we curate a large and diverse dataset of URLs from multiple online sources. This is followed by a discussion on our dataset construction and cleaning process to ensure high-quality data for our experiments.

Next, we describe the machine learning models used, including the Support Vector Classifier (SVC) and Stochastic Gradient Descent (SGD) Classifier. Their respective configurations, hyperparameters, and reasons for selection are provided. We then discuss the feature representation and tokenization strategies for the input data. Three types of inputs and three tokenization methods are utilized for this purpose.

Following that, we describe the experimental setup, including the evaluation metrics employed, which consist of precision, recall, and F1 and F2 scores. Finally, we discuss the data analysis phase, in which the results of our experiments are evaluated and compared.

Data Collection

The dataset used in this research has been curated from multiple online sources [8–14], providing a diverse set of URLs including benign, malicious, and adult. The primary

source for benign URLs is the URL Classification Dataset [DMOZ] [8], while the primary source for malware URLs is URLhaus [14], the URLhaus dataset of URLs is updated over time; we downloaded and used the dataset with the last update of first of March 2022. After collecting the URLs, a crawling process is initiated for each URL to gather the raw HTML content, for the crawling process, we used the "OWler" which is a crawler developed by Dinzinger et al. [submitted to OSSYM2023]. This content is used for the comparison of machine learning model performance when raw HTML content, parsed HTML content, or only the URL is used as input. Post-collection, the raw HTML content is parsed to extract structured content, which is stored as a distinct field for each URL in the dataset. Here we faced a problem where we had some URLs that were not working anymore, in this case, we just ignored any non-working URL. Each URL is further labeled with a main label and a subclass label, providing 3 and 20 unique labels, respectively.

Dataset Construction

To ensure data quality and relevance, the dataset undergoes a cleaning process, which includes eliminating duplicates where we removed around two thousand duplicate URLs, most of which were malicious URLs. Then the URLs with empty content were also removed, in this step, we found that 23 URLs had no content when they were crawled, these 23 URLs were removed from our dataset. The cleaning procedure ensures that the dataset is reliable and can be effectively utilized for the experiments planned in this study. The count of each category and sub-category can be found in Table 1.

Table 1: Count of each category and subclass

Main Label	Subclass	Count	Total
Adult	Adult	4424	4424
Malicious	Spam	830	22949
	Phishing	3734	
	Defacement	4004	
	Malware	14381	
Benign	Society	22010	88628
	Arts	15073	
	Privacy Policy	10575	
	Science	9408	
	Computers	4828	
	Games	4270	
	Recreation	4231	
	Reference	3707	
	Business	3641	
	Sports	2986	
	Kids	2392	
	Health	2110	
	Shopping	1572	
	Home	1475	
	News	350	

Machine Learning Models

Two models have been employed in this study, Support Vector Classifier (SVC) and Stochastic Gradient Descent (SGD) Classifier. The SVC model [15] uses a linear kernel, allowing it to scale well to large datasets, thanks to its implementation in terms of liblinear [16] rather than libsvm [17]. The SGD Classifier is a linear classifier optimized using stochastic gradient descent, making it particularly useful for large datasets due to its suitability for online or mini-batch learning settings.

Both models' hyperparameters were mostly set to the default values. For SGD, we set the loss function to 'hinge', the regularization penalty was set to 'l2' with an alpha of 0.0001. For SVC, the penalty was set to 'l2' with a loss of 'squared hinge', both models were fit with an intercept and the maximum number of iterations for both was set to 1000, the only hyperparameter that was set to a non-default value was class_weight which we set to 'balanced' in order to mitigate the unbalanced classes.

Both SVC and SGD classifiers are linear models which make them well suited for large scale feature datasets like ours. In terms of computational cost and memory usage, they are efficient and this is crucial in handling our dataset of 116 thousand URLs.

Feature Representation and Tokenization

Feature representation in this study encompasses three types of input: URLs only, raw HTML content, and parsed HTML content. Each input type possesses its own unique strengths and weaknesses for the classification task at hand. To explore the impact of tokenization methods and levels, we employed character-level, word-level, and Byte Pair Encoding (BPE) [1] techniques. In particular, we chose a window size of (1,3) for character-level and word-level tokenization. This decision was motivated by the desire to capture both local and contextual information within the text. By considering 1-grams, 2-grams, and 3-grams simultaneously, we aimed to extract fine-grained details as well as broader contextual patterns, striking a balance between granularity and computational complexity.

Experimental Setup

The experimental setup includes a total of 36 experiments, each designed to investigate a specific combination of models, input types, and tokenization methods. We constructed the settings of the 36 experiments to cover all the possible combinations of the following aspects: algorithms used for classification (Stochastic Gradient Descent and Support Vector Classification), tokenization methods (TF-IDF and Byte Pair Encoding), types of input data (URL, Content, and HTML), labels (Main label and Subclass), and the levels of n-grams (character-level and word-level).

For each experiment, the dataset was split into a training set and a test set at a ratio of 70%, 30%, respectively, resulting in 81,200 URLs for training and 34,801 URLs for testing. This split provides enough data for training while

still reserving a sizable portion for validation, which ensures the reliability of the experiment's results.

The performance of the experiments is evaluated based on precision, recall, and F1 and F2 scores. The F1-score is the harmonic mean of precision and recall, while the F2-score is more sensitive to recall. We chose to include the F2-score in our evaluation metrics because we prioritize the recognition of illegal or harmful web pages. In such cases, high recall (i.e., reducing the number of false negatives) is more important than precision, as missing such pages could lead to more severe consequences than falsely identifying a harmless page as harmful.

Both the F1 and F2 scores are calculated for each class and then averaged to produce macro-averaged scores, thus ensuring an unbiased measure across the classes.

RESULTS

This section provides an evaluation of the 36 conducted experiments, employing the F2 macro score as the primary criterion for comparison. Various elements were analyzed, including input types (URL, content, and HTML), tokenization methodologies (TFIDF and BPE), as well as machine learning algorithms (Stochastic Gradient Descent - SGD and Support Vector Classifier - SVC).

Based on our evaluation of model performance with various types of inputs, we consistently found that the use of URL input leads to superior model outcomes compared to those utilizing content or HTML input. This is evident from the high F2 scores achieved when the target output is the main label. To illustrate, the main label classification yielded an F2 score of 0.94 for SVC and 0.92 for SGD with URL input. However, in the case of subclass classification, the model which leverages SVC algorithm with content input was superior, achieving an optimal F2 score of 0.64. Despite this, URL input maintained its efficiency edge in terms of prediction and training time, outshining both content and HTML inputs, as shown in Figure 1.

Analyzing the confusion matrix of the best performing model, as shown in Figure 2, gives us insights into the model's performance across the different classes: 'Adult', 'Benign', and 'Malicious'. The 'Adult' class had an accuracy of 88%, with 12% of instances being misclassified as 'Benign', while no instances were misclassified as 'Malicious'. The 'Benign' class showcased an impressive accuracy of 99%, with only 1% of instances incorrectly identified as 'Malicious'. For the 'Malicious' class, 92% of instances were correctly classified, with 8% being wrongly classified as 'Benign'. No 'Malicious' instances were misclassified as 'Adult'. This suggests that the classifier performs exceptionally well for the 'Benign' and 'Malicious' classes, with room for improvement in the detection of 'Adult' content.

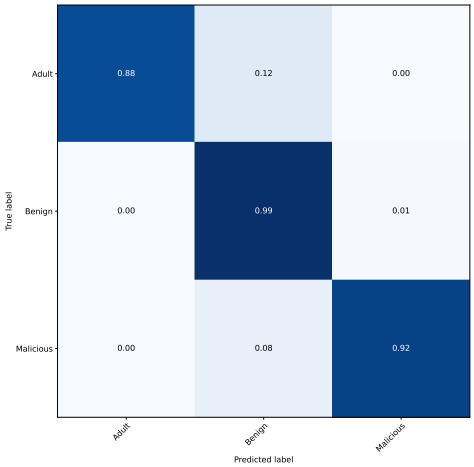


Figure 2: Confusion Matrix of the Best Model

Our analysis of the top 20 most important features, illustrated in Figure 3, offers compelling insights into how the classifier makes its decisions. With TFIDF tokenization at 1, 2, and 3-grams word level, and a trained Random Forest model, the term "video" emerged as the most significant feature, followed by "com video", "HTTP", "www", "https", and "com". The prominence of the term "video" in the feature importance ranking suggests that URLs containing this term are more likely to be classified as adult webpages. Similarly, the presence of "zip" amongst the important features indicates the URL's probable classification as a malicious webpage, potentially hosting malware. The other 14 features do not display such high importance values. While these findings do suggest a potential bias of the classifier towards URLs containing these key terms, it also underscores the model's ability to discern patterns and relationships between specific words and webpage classification. It is crucial, however, to approach this interpretation with caution, as it might not always be the case, and further research is needed to assert these relationships conclusively.

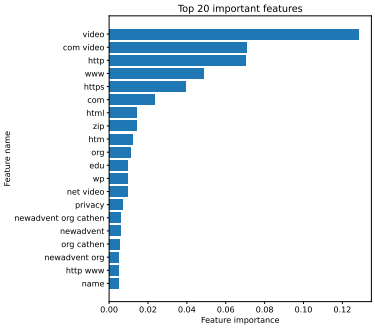
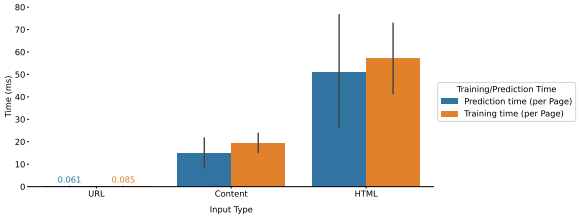
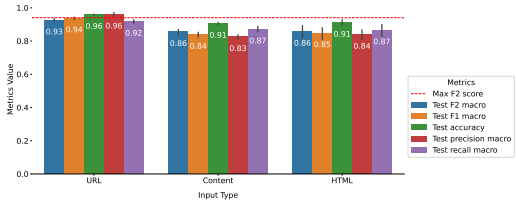


Figure 3: Top 20 Features

Switching our focus to tokenization methodologies, TFIDF generally outperforms BPE. This is evident in main label classification where TFIDF surpasses BPE across all inputs and algorithms, most notably reaching an F2 score of 0.94 with SVC on URL input. Predictive efficiency and



(a) Train/Test time per URL for different Input types (Output: Main Label)



(b) Model performance metrics for different Input types (Output: Main Label)

Figure 1: Performance of the best model

Table 2: Highlights of the results. Note: The variance of all measured values is very small, approaching 0.

algorithm	tokenizer	input	output	F2	prediction time(ms)	fit time(ms)	token level	accuracy	precision	recall
SVC	TFIDF	url	main label	0.94	0.05	0.10	char	0.96	0.96	0.94
SVC	BPE	url	main label	0.93	0.12	0.13	-	0.96	0.95	0.93
SGD	TFIDF	url	main label	0.92	0.05	0.07	char	0.96	0.97	0.91
SVC	TFIDF	html	main label	0.90	13.79	43.58	word	0.93	0.87	0.92
SVC	TFIDF	content	main label	0.87	13.95	14.24	char	0.91	0.83	0.90
SVC	BPE	content	subclass	0.64	25.76	26.85	-	0.74	0.61	0.66
SVC	TFIDF	url	subclass	0.62	0.05	0.26	char	0.72	0.59	0.64
SGD	TFIDF	html	subclass	0.48	51.66	45.00	char	0.60	0.48	0.51

training times also align with these results, with TFIDF maintaining a lead. However, the gap between TFIDF and BPE narrows down in the subclass classification. An instance of this can be observed with SVC, which delivers higher F2 scores with TFIDF on URL and content inputs, but sees a slight improvement with BPE on HTML input.

Delving into the comparison between SGD and SVC as machine learning models, SVC frequently yields superior results in terms of F2 macro scores. A manifestation of this can be seen in SVC's highest F2 score of 0.94 with URL input and TFIDF tokenization for main label classification, and 0.63 for subclass classification under the same conditions. SGD, however, falls short with highest scores of 0.92 and 0.56 respectively. On the other hand, SGD's prediction and training times are consistently faster than those of SVC. For results of more experiments see Table 2, it should be noted that Table 2 only showcases selected key outcomes from the total of 36 experiments we conducted.

To sum up, our experiments show that the combination of SVC algorithm and TFIDF tokenization applied to URL input yields the highest F2 macro scores It's noteworthy that this optimal configuration does not invariably ensure the most efficient prediction and training times. Lastly, the n-gram level's influence (word or char) seems less impactful in these experiments, thereby emphasizing the importance of appropriate feature selection and algorithm choice for machine learning tasks in webpage classification.

DISCUSSION

The results of our study offer several noteworthy insights into the process of webpage classification, particularly focusing on the selection of features and machine learning algorithms. Our findings primarily highlight the potential of URL inputs, the TFIDF tokenization method, and the SVC algorithm to yield high classification performance. These results extend and deepen the understanding of webpage classification, presenting potential guidelines for feature and algorithm selection in this field.

The observed superior performance of URL inputs over content and HTML inputs aligns with the previous research asserting the high information value embedded in URLs. This finding builds upon the studies by Abdallah and de La Iglesia [5], and Kan and Thi [4], who have also leveraged URL inputs for webpage classification. Notably, our study expands on these works by illustrating that URL inputs not only yield high accuracy but also ensure superior efficiency in terms of prediction and training times.

Similarly, our analysis of tokenization methods adds to the current body of literature. The observed dominance of TFIDF over BPE in most scenarios is an important contribution, especially when considering the main label classification. Although the performance difference in subclass classification is less pronounced, the findings still shed light on the potential implications of tokenization methods for webpage classification, encouraging future researchers to consider these aspects when designing their classification models.

In terms of machine learning algorithms, the superior performance of SVC over SGD in our study presents an interesting point for discussion. While previous research has demonstrated the utility of a range of machine learning algorithms for webpage classification, including XGBoost [6], our findings point out the potential advantages of SVC, particularly when combined with TFIDF tokenization and URL input. This, however, does not discount the potential util-

ity of SGD, which displayed competitive results and higher efficiency in terms of prediction and training times.

However, it is important to acknowledge the limitations of our study. Our focus was restricted to a limited set of features, tokenization techniques, and machine learning algorithms. This presents an expansive opportunity for future research to explore a wider range of methods and techniques that could potentially enhance the scope and applicability of webpage classification tasks.

Future research in this area could consider incorporating additional features, tokenization methods, or machine learning algorithms. Additionally, the impact of different preprocessing steps, feature selection methods, or hyperparameter tuning approaches could be investigated.

CONCLUSION

In conclusion, this research introduces a comprehensive dataset of 116,000 URLs, providing a substantial resource for future research in the field of webpage classification. Through comprehensive analysis, it became evident that URLs represent a highly valuable input source, consistently yielding superior model outcomes compared to other inputs such as HTML content.

The study's findings revealed that the Support Vector Classifier (SVC), in conjunction with TFIDF tokenization and URL input, yielded the highest F2 macro scores. Although this optimal combination does not invariably ensure the most efficient prediction and training times, it does highlight the importance of careful feature selection and algorithm choice for tasks in webpage classification.

Moreover, tokenization significantly impacts performance, underscoring the importance of feature representation. Results favored TFIDF over BPE in most cases, with n-gram level playing a minor role. Although SGD and SVC showed similar accuracy, SVC outperformed in F2 macro scores, indicating its aptness for this task.

This study enhances web crawling applications by identifying optimal techniques for feature representation and model choice. It paves the way for future work in webpage classification, providing key insights and a rich dataset for continued research.

ACKNOWLEDGEMENTS

This work is part of the OpenWebSearch.eu project, funded by the EU under the GA 101070014, and part of the CAROLL project, funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01|S20049.

REFERENCES

[1] Philip Gage. A new algorithm for data compression. *C Users Journal*, 12(2):23–38, 1994.

[2] Michael Granitzer, Stefan Voigt, Noor Afshan Fathima, Martin Golasowski, Christian Guetl, Tobias Hecking, Gijs Hendriksen, Djoerd Hiemstra, Jan Martinovič, Jelena Mitrović, et al. Impact and development of an open web index for open web search. *Journal of the Association for Information Science and Technology*, 2023.

[3] Inma Hernández, Carlos R Rivero, David Ruiz, and Rafael Corchuelo. A statistical approach to url-based web page clustering. In *Proceedings of the 21st International Conference on World Wide Web*, pages 525–526, 2012.

[4] Min-Yen Kan and Hoang Oanh Nguyen Thi. Fast webpage classification using url features. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 325–326, 2005.

[5] Tarek Amr Abdallah and Beatriz de La Iglesia. Url-based web page classification: With n-gram language models. In *Knowledge Discovery, Knowledge Engineering and Knowledge Management: 6th International Joint Conference, IC3K 2014, Rome, Italy, October 21-24, 2014, Revised Selected Papers 6*, pages 19–33. Springer, 2015.

[6] Ali Aljofey, Qingshan Jiang, Abdur Rasool, Hui Chen, Wenyin Liu, Qiang Qu, and Yang Wang. An effective detection approach for phishing websites using url and html features. *Scientific Reports*, 12(1):8842, 2022.

[7] Hung Le, Quang Pham, Doyen Sahoo, and Steven CH Hoi. Urlnet: Learning a url representation with deep learning for malicious url detection. *arXiv preprint arXiv:1802.03162*, 2018.

[8] Dmoz dataset, Accessed: 2022-11-10. `https://www.kaggle.com/datasets/shawon10/url-classification-dataset-dmoz`.

[9] Malicious dataset, Accessed: 2022-11-10. `https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset`.

[10] Spam urls classification dataset, Accessed: 2022-11-10. `https://www.kaggle.com/datasets/shivamb/spam-url-prediction`.

[11] Spam urls - spam 404, Accessed: 2022-11-10. `https://raw.githubusercontent.com/Spam404/lists/master/main-blacklist.txt`.

[12] Spam urls - matomo.org, Accessed: 2022-11-10. `https://raw.githubusercontent.com/matomo-org/referrer-spam-blacklist/master/spammers.txt`.

[13] Maps policies dataset v1.0, Accessed: 2022-11-10. `https://www.usableprivacy.org/static/data/MAPS_Policies_Dataset_v1.0.zip`.

[14] Urlhaus, Accessed: 2023-3-1. `https://urlhaus.abuse.ch/`.

[15] Corinna Cortes and Vladimir Vapnik. Support vector machine. *Machine learning*, 20(3):273–297, 1995.

[16] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *J. Mach. Learn. Res.*, 9:1871–1874, jun 2008.

[17] Chih-Chung Chang and Chih-Jen Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.