DETECTION OF MALICIOUS WEBSITES USING MACHINE LEARNING

A.Haseena[1]., Shaik Tasleema Tabassum[2]., V.Sai Tejasri[3]., N.Vyshali[4] ., V.Maneesha[5]

1 Assistant Professor, Department of ECE., Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India (✉ haseenamtech007@gmail.com)

2, 3, 4, 5 B.Tech ECE, (19RG1A0450, 19RG1A0460, 19RG1A0437, 19RG1A0459),

Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

Abstract

Common methods for identifying harmful websites rely on blacklists, which are not comprehensive and cannot generalise to new dangerous sites. Automatically identifying freshly visited dangerous websites is an important step in reducing exposure to such attacks. In this research, we investigated 10 machine learning models' ability to generalise across datasets by classifying dangerous websites based on lexical properties. To be more precise, we performed a cross-dataset study after training, validating, and testing these models on many datasets. Our research shows that K-Nearest Neighbour is the only model that provide excellent results across all datasets tested. Across all metrics and datasets, other models, such as Random Forest, Decision Trees, Logistic Regression, and Support Vector Machines, outperform a basic model of always classifying every link as harmful. Further, we could not discover any lexical characteristics that generalised well across models and datasets. Cybersecurity experts and academics should find this study interesting since it might help inform the development of practical detection systems or inspire new lines of inquiry in the field.

Keywords—

dangerous URLs, artificial intelligence, and lexical characteristics.

INTRODUCTION

In recent years, internet use has increased as more and more people and organisations find it useful for their own purposes as well as those of their businesses. Sadly, some groups are using the internet's widespread availability to do criminal acts. Malicious websites are a popular vector for such attacks. A harmful website is one that tries to infect visitors' computers with malware in order to steal their personal information or get access to sensitive data like passwords [1–3]. Because of this, our study focuses on the problem of identifying bad URLs, which has garnered a lot of attention in the world of cybersecurity. A blacklist is a frequent tool for finding harmful URLs. Although blocking access to a certain URL has proven successful, the dynamic nature of the Internet implies that blacklists are insufficient protection against this kind of assault. Because machine learning methods may be used to discover harmful websites even if they have never been seen before, unlike a blacklist, they have been advocated as an effective tool in monitoring rogue URLs [4]. Machine learning (ML) refers to the study of algorithms that can learn and become better on their own [5]. ML models provide superior insights into URL classification [6 -7] due to their ability to comprehend the underlying lexical structure of URLs.

CONTEXT AND HISTORICAL WORK

Researchers have suggested a number of different methods for identifying fraudulent websites, including machine learning, data mining, and even deep learning. The effectiveness of these methods relies heavily on the quality and combination of important characteristic elements of web pages [8]. These variables may include data about network traffic, content characteristics, lexical properties of URLs, and data about the domain name system (DNS). Inconvenient for real-time systems [7, 11], attribute extraction can be time-consuming and expensive [9] if it requires downloading entire web pages or [10] if it requires querying multiple DNS servers and Internet service providers to obtain enrichment data such as geo-location, registration records, and network information. Previous works have investigated the use of URL lexical features alone and shown that URL features alone can produce an accurate means of detecting malicious webpage in real time systems [7, 9, 13], despite the challenges of using non-lexical features to detect malicious URLs in real time despite their high accuracy values.

The performance reported in the literature [7, 9, 13–17] for systems that simply utilise lexical-based characteristics varies greatly depending on the model and dataset used. Specifically, the authors of [14] compared three supervised machine learning models (k-nearest neighbour, support vector machine (SVM), and naïve bayes classifier) with two unsupervised machine learning models (k-means and affinity propagation). The study concluded that supervised models performed marginally better than their unsupervised machine learning counterparts. Also, the data set can be too big. As an example, in [16], numerous machine learning models were employed in conjunction with association rule mining to determine if a given URL was harmful or not depending on the data extracted from the URL itself. The authors used the Synthetic Minority Over-Sampling Method to address the class imbalance and limited size of the dataset (SMOTE). Comparing the models' performance before

and after class balancing demonstrates that most models improved greatly after class balancing and the accompanying increase in dataset size.

METHODOLOGY

Data collecting, data pre-processing, lexical feature engineering, machine learning modelling, and cross datasets analysis are the five key sub-problems that may be broken out from the core research subject.

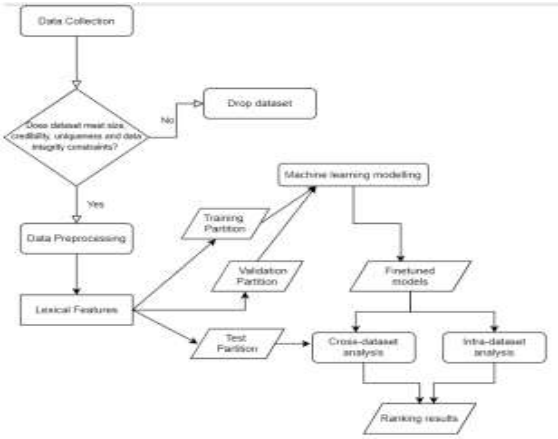

Fig. 1 The interconnection between sub-problems.

Collecting Information

For this reason, we needed to compile the datasets upon which our analysis relied. These are the individual phases:

1) Conducting a manual search using internet datasets repositories like datasetsearch.google.com and Kaggle.com to locate potential datasets for analysis.

2) Selectively removing information from a dataset based on criteria including its size, originality, and reliability. The threshold for uniqueness in this study is a maximum of 20% percent overlap with the other datasets. The reliability of a dataset may be judged by the transparency with which its curation process is described. This would provide a prioritised list of data sets for auditing.

3)we validate the accuracy of the data by double-checking the names of 100 randomly selected rows from each dataset. As a result of this human-verified 80% accuracy threshold, any datasets that did not pass were removed. Processes Performed Before Viewing Data.

4)Datasets have to be standardised so that they all had the same structure, consisting of a URL (a string) and a label (benign or malicious). At this point, we had to write some Python scripts in order to standardise the dataset we were working with. In the end, these scripts will have ensured that all datasets are uniform and ready for analysis. Engineering of Lexical Features.

In order to do this analysis, we needed to generate values for the feature space based on the linguistic characteristics of URLs.

The following were the stages of lexical feature engineering:

1) Investigating published works to collect lexical characteristics essential to our study.

2) Ordering these characteristics according to their widespread use, significance to the canon, and originality. Thirdly, using our expertise in the field to design a wholly original feature. Fourth, the engineered characteristics may be retrieved by writing python programmes that accept the standard dataset as input.

3)Training data (34%), validation data (33%), and test data (33%).

Methods of Machine Learning Modelling

Our whole investigation hinged on these findings. It entailed transforming the standard dataset into useful models. The specific issue here was that:

1) Identifying and narrowing down machine learning models used in the literature for malicious website identification and related tasks.

2) Narrowing the pool of potential machine models down to ten using empirical performance metrics from both the literature on malicious website identification and other popular tasks.

3)Choosing Metrics for Model Training and Validation.

4) Constructing machine learning models, training them on the training partition, and then validating them using hyperparameter optimization on the validation set to guarantee they are optimally adjusted for the job at hand.

5) Putting these optimised models away for use in research involving several data sets.

CONCLUSIONS AND RESULTS

We compiled over 2 million URLs into 16 datasets with sizes ranging from 20,000 to 600,000. To guarantee that each URL is distinct, we eliminated those that were identical. As a result of the duplication elimination method, the dataset only contains URLs that are lexically similar but ultimately link to separate sites, such as www.google.com, https://www.google.com, and google.com. We didn't remove these URLs since they provide lexical diversity that contributes to the system's overall resilience. Processes Performed Before Viewing Data.

The representations vary amongst the datasets since they came from various places. While some data sets may identify dangerous URLs with a 1, others may use a text indicating that the URL is malicious. We normalised all 16 datasets to include a URL column with the whole URL and a label of 0 for safe sites and 1 for malicious ones. The malicious percentage ranged from around 30% to 35% across the different data sets. Engineering of Lexical Features. We used literature data to generate 78 lexical characteristics, including data on the length of the hostname, the top-level domain, the number of routes in the URL, and the number of unusual characters in the hostname and URL. Furthermore, 300 word2vec-based features were available, all of which were built on the ground-breaking work presented in [7]. On top of that, we expanded upon [7]'s work by suggesting two new features: the benign score and the mischievous score.

Instead of using a blacklist-based n-gram model as they did, we developed separate "language models" to identify malicious and safe URLs. The goal of language modelling (LM) is to predict how likely it is that a certain sequence of words will appear in a sentence in a given language [21]. In this study, we looked at the likelihood of a certain string of characters appearing in a URL, depending on whether or not the URL is safe to use. The dangerous URL language model would provide a greater probability to the URL string if it were malicious, whereas the benign URL language model would assign a lower probability if the URL was benign. If the URL is harmless, the opposite is true. These two rankings should be fairly indicative of a URL's potential for harmful activity. After extensive feature engineering, we settled on a set of 380 characteristics that we thought would be particularly good at predicting whether a URL was dangerous or not.

The Modelling Strategies of Machine Learning

There are eight supervised models and two unsupervised models in our final models, for a total of six sets of classifiers. models based on trees (Decision Tree, Random Forest, Categorical Boosting), models based on networks of neighbours (K-Nearest Neighbours), models based on neural networks (Feed Forward Neural Network), and statistical models (Naive Bayes) - Models that do not need human supervision (such as the K-Means and the GMM).

Area under the Receiver Operating Curve (AUC-ROC), Recall, Precision, F1, and Accuracy were the five measures we picked for our comparison study (ACC). There were 6, 5, and 5 distinct datasets used for training, validating, and testing. In addition to training and validating the model, various steps such as feature selection, feature scaling, dimensionality reduction, and hyperparameter tuning were performed. All 10 models indicated that word2vec based features were not predictive of maliciousness when it came to feature selection. We attribute this to the fact that most word2vec models are data set unique. This renders them less suitable for use in a combined dataset analysis. But eight out of ten models found our innovative traits to be relevant in the prediction, indicating their evident worth. Last but not least, across all datasets, no cluster of attributes was consistently selected as the best predictor by any of the models.

Analysing Multiple and Individual Datasets

The data from the study of a single dataset are shown in Table 1. The average findings from the cross-dataset analysis are shown in Table 2. Using the average rankings acquired for each measure across all five test datasets (DS1 - 5) in the combined dataset analysis, Table 3 displays the models' average rank performance. If your model outperforms the baseline model on all measures, we call this "rank performance," or "RNK," and it will determine where your model stands among other ML models. As a starting point, we opted for a simplistic classifier that incorrectly labels all links as harmful. Finally, we round the mean rank performance up or down to get the average rank performance.

TABLE I. TABLE OF METRIC SCORES FOR SINGLE DATA SET ANALYSIS

F1	*AUC-ROC*	*ACC*	*REC*	*PCSN*
0.78	0.83	0.85	0.73	0.84

TABLE II. TABLE OF AVERAGE METRIC SCORES ACROSS ALL DATASETS FOR CROSS_DATASET ANALYSIS

	METRICS				
DATASET	*F1*	*AUC-ROC*	*ACC*	*REC*	*PCSN*
DS 1	0.39	0.58	0.52	0.68	0.38
DS 2	0.39	0.55	0.55	0.43	0.71
DS 3	0.39	0.55	0.55	0.43	0.71
DS 4	0.39	0.55	0.55	0.44	0.71
DS 5	0.39	0.55	0.55	0.44	0.70

TABLE III. TABLE OF AVERAGE MODEL RANK PERFORMANCE ACROSS DATASETS

	DATASETS					
MODELS	*DS 1*	*DS 2*	*DS 3*	*DS 4*	*DS 5*	*RNK*
KNN	2	1	2	1	1	1
SVM	3	5	4	4	4	4
RF	10	4	4	4	4	5
DT	5	7	6	6	6	6
LR	6	6	7	6	6	6
NB/CB/ GMM/ KMEANS/ FFNN	10	10	10	10	10	10

Even though we only modified the models for the cross-dataset study, Table I shows that their performance on the single-dataset analysis was satisfactory. In other words, any subpar results from the cross-dataset study could not be attributable to the decisions we made in our models. As can be seen in Table II, when a cross-dataset analysis is undertaken, the models' average performance quickly declines, suggesting that they have trouble replicating their success.

CONCLUSION

Our findings indicate that KNN is the most generalizable model for using lexical characteristics across different datasets. Models such as Support Vector Machines, Random Forests, Logistic Regression, and Decision Trees are anticipated to outperform the current state of the art, which assumes that each and every connection is harmful. As a result, it makes sense to use tree-based models or linear models for analysing a single dataset in order to identify dangerous relationships. Our research, however, demonstrates that their effectiveness diminishes when applied to many datasets instead of just one. Since there is no assurance that their stated performance will hold up when tested on links that were not part of the initial training/validation set, caution should be used before trusting their reported performance. Our findings suggest that KNN is the only model that consistently perform well across data sets. This might be due to the dominance of the closest neighbour concept. To find out why, further in-depth empirical research must be conducted. We also could not discover any proof that a particular collection of lexical traits is transferable between data sets. Nonetheless, our processing options could have had a role in this. Consequently, this might be tested in subsequent studies by using other processing options. Finally, our use of lexical features in this study means there is no proof that the analysis will apply to other popular characteristics, such as content-based or DNS-based features. Therefore, this study might be expanded in the future to include more feature sets, either alone or in combination.

REFERENCES

[1] B. Eshete, A. Villafuerte and K. Teklemariam, "Malicious Website Detection: Effectiveness and Efficiency Issues", 2011 First Sysco Workshop, 2011. Available: 10.1109/syssec.2011.9.

[2] A. Ali Ahmed, "Malicious Website Detection: A Review", Journal of Forensic Sciences & Criminal Investigation, vol. 7, no. 3, 2018. Available: 10.19080/jfsci.2018.07.555712.

[3] Norton LifeLock, "Norton," Norton.com, 2019. https://us.norton.com/internetsecurity-malware-what-are-maliciouswebsites.html.

[4] F. Vanhoenshoven, G. Nápoles, R. Falcon, K. Vanhoof and M. Köppen, "Detecting Malicious URLs using Machine Learning Techniques," 2016 IEEE Symposium Series on Computational Intelligence (SSCI), Athens, Greece, 2016, pp. 1-8, DOI: 10.1109/SSCI.2016.7850079.

[5] M. Jordan and T. Mitchell, "Machine learning: Trends, Perspectives, and Prospects", Science, vol. 349, no. 6245, pp. 255-260, 2015. Available:10.1126/science.aaa8415

[6] A. S. Manjeri, K. R., A. M.N.V., and P. C. Nair, "A Machine Learning Approach for Detecting Malicious Websites using URL Features," 2019 3rd International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 555-561, DOI: 10.1109/ICECA.2019.8821879.

[7] Q. T. Hai and S. O. Hwang, "Detection of Malicious URLs Based on Word Vector Representation and Ngram," Journal of Intelligent & Fuzzy Systems, vol. 35, no. 6, pp. 5889–5900, Dec. 2018, doi: 10.3233/jifs169831.

[8] D. SAHOO, C. LIU, and S. HOI, "Malicious URL Detection using Machine Learning: A Survey", Arxiv.org, 2021. [Online]. Available: https://arxiv.org/pdf/1701.07179.pdf.

[9] A. Y. Daeef, R. B. Ahmad, Y. Yacob, and N. Y. Phing, "Wide Scope and Fast Websites Phishing Detection using URLs Lexical Features," 2016 3rd International Conference on Electronic Design (ICED), Phuket, Thailand, 2016, pp. 410-415, doi: 10.1109/ICED.2016.7804679.

[10] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A Survey on Malicious Domains Detection through DNS Data Analysis", ACM Computing Surveys, vol. 51, no. 4, pp. 1-36, 2018. Available: 10.1145/3191329.

[11] K. Rieck, T. Krueger, and A. Dewald, "Cujo: Efficient detection and Prevention of Drive-by-download Attacks," in Annual Computer Security Applications Conference (ACSAC), 2010, pp. 31–39.

[12] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond Blacklists: Learning to detect Malicious Websites from Suspicious URLs." Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '09, 2009, DOI: 10.1145/1557019.1557153.

[13] A. Joshi, L. Lloyd, P. Paul Westin, and S. Seethapathy, "Using Lexical Features for Malicious URL Detection -- A Machine Learning Approach", 2019.

[14] H. Kazemian and S. Ahmed, "Comparisons of Machine Learning Techniques for Detecting Malicious Web Pages", Expert Systems with Applications, vol. 42, no. 3, pp. 1166-1177, 2015. Available: 10.1016/j.eswa.2014.08.046.

[15] A. S. Manjeri, K. R., A. M.N.V., and P. C. Nair, "A Machine Learning Approach for Detecting Malicious Websites using URL Features," 2019 3rd International Conference on Electronics, Communication, and Aerospace Technology (ICECA), Coimbatore, India, 2019, pp. 555-561, DOI: 10.1109/ICECA.2019.8821879.

[16] M. Darling, G. Heileman, G. Gressel, A. Ashok and P. Poornachandran, "A Lexical Approach for Classifying Malicious URLs",

