Detection of Malicious URLs using Machine Learning based on Lexical Features

Prabodha Abeynayake
Department of Computer Engineering
University of Sri Jayewardenepura
Rathmalana, Sri Lanka
pyabeynayake@gmail.com

Udaya Wijenayake
Department of Computer Engineering
University of Sri Jayewardenepura
Rathmalana, Sri Lanka
udayaw@sjp.ac.lk

***Abstract*—As the digital world evolves, the risk of valuable information being exposed to unauthorized parties is increasing. One common vulnerability is the use of malicious Uniform Resource Locator (URL), which are fraudulent links spread across various platforms such as social media and emails. Traditional methods of identifying these URLs, such as blacklisting and heuristic search, rely heavily on syntax or keyword matching but struggle to keep up with the evolving tactics of cyber attackers. Hence, this paper proposes a solution for detecting malicious URLs and their types based on lexical features. Lexical features in a URL refer to the components that convey semantic and lexical meaning. These can include domain names, path lengths, special characters, and other elements that can be analyzed for patterns or anomalies. In our proposed method, we use 23 different lexical features that focus on the semantic and lexical meaning of the URLs. An Exploratory Data Analysis (EDA) is used to filter the most important lexical features that effectively contribute to predictions. With these carefully curated features, we address the problem as a multi-classification task, aiming to assess the performance of three distinct classifiers: Random Forest, which currently stands as the domain's best solution and a pure bagging technique, as well as XG Boost and Light GBM, both of which utilize boosting techniques. With the proposed method, we could achieve over 93% accuracy for all three classifiers while Random Forest achieving the best performance.**

***Index Terms*—malicious URLs, lexical features, cybercrime, classifiers, machine learning**

I. Introduction

Uniform Resource Locator (URL) is a mechanism of accessing a source published on the web. With human-readable format, URLs are much preferred by the general users rather than their respective IP (Internet Protocol) addresses. As technology is evolving and the increasing activities of cybercrimes, the security of the information on the networks has been a great problem. Symantec's 2019 Internet Security Report [1] states the various cybercrimes, attacks and threats as well as the common mechanisms used by the attackers to lure naïve users into fraudulent activities such as malware, formjacking and ransomware. An eminent mechanism that ranked among the top ten tactics used by the attackers is luring the users to click on a malicious URL that would make them vulnerable to loss of sensitive and confidential data and systems.

There are two fundamental components of the URL, which are the protocol identifier that indicates which protocol to use, and the resource name that indicates the IP address or the domain name where the resource is located [2]. Hence, the URL has a specific format, the attackers usually tend to alter the format by adding and removing other components to the URL to deceive the users and then spread the malicious URLs among users in a conniving manner [1].

As identifying malicious URL was extremely crucial to prevent cybercrimes, multiple methods were developed such as blacklisting services and heuristic classification. In the Blacklisting method, it contains a database comprising previously identified malicious URLs. Upon encountering a new URL, the system performs a database lookup, testing the new URL against every entry in the Blacklist. When a new malicious URL is identified, it is promptly added to the Blacklist [3]. The heuristic approach is an improved version of the blacklisting approach. In this method, signatures are employed to compare and assess the correlation between a new URL and the signatures of known malicious URLs [4].

These techniques have been effective to a certain extent in preventing users from accessing malicious URLs. However, as the volume of URLs on the web continues to grow, accompanied by the daily influx of newly generated URLs, the efficiency of these methods has diminished. As conventional methods are not enough to cope with evolving technology and tactics of cyber attackers, new methods of addressing the problem have been explored from a Machine Learning standpoint.

In this paper, we propose a method that uses 23 lexical features which focus on the semantic and lexical meaning of the URLs. We leverage machine learning algorithms to classify URLs as either benign or malicious. Furthermore, if a URL is determined to be malicious, we identify its specific malicious type solely based on the lexical features extracted from the URL string. This study also provides a proper feature extraction method using an Exploratory Data Analysis (EDA) to determine the features that effectively contribute to the prediction of a URL as malicious or benign. The research has also made an effort to explore the performance between pure bagging and boosting techniques.

The word cloud in Figure 1 displays the most common keywords found in malicious URLs as a wordcloud, providing insight into the prevalent terms used in these URLs, which is valuable for the lexical feature-based classification approach.

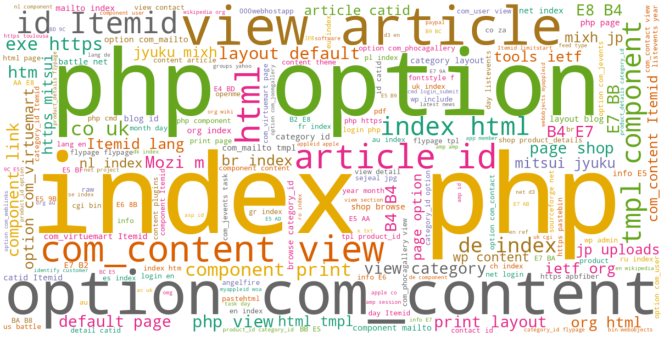

Fig. 1. Malicious URL Wordcloud's Word frequency visualized, with larger text denoting higher occurrence

The organization of this paper is as follows: Section II discusses related works in the field. Section III presents the methodology. In Section IV, results and discussion are provided. Section V provides the concluding remarks.

II. Related Work

In the realm of malicious URL detection, prior research has embraced various approaches, including lexical and composite feature sets, binary and multi-classification methods, and a diverse range of machine learning techniques. Notably, some studies have also incorporated mathematical functions, statistical analyses, and gradient optimization methods, contributing to the diversity of strategies applied to this critical challenge.

Naveen et al. [5] and Xuan et al. [6] utilized mixed feature sets for binary URL classification. In their work, Naveen et al. [5] proposed an early methodology of their work which employed 18 features, including lexical, third-party, geo-ranking, network, and URL behavioral attributes, for supervised binary classification. Meanwhile, Xuan et al. [6] presented a binary classifier with Support Vector Machine (SVM) and Random Forest (RF) to improve the detection of malicious URL with machine learning and Big Data techniques. For the experiments, they used a dataset consisting of 470,000 different URLs. According to their experiments, Random Forest with 100 trees had the best performance with 96.28% accuracy, outperforming the SVM by nearly 5%.

Nair et al. [7] surveyed machine learning techniques and algorithms for malicious URL detection, providing a comprehensive exploration of machine learning techniques, feature representations, and learning algorithms essential for effective malicious URL detection, highlighting its pivotal role in strengthening cybersecurity applications.

Cui et al. [8] and Zhao et al. [9] are among the researchers who have incorporated slightly diverse strategies compared to others in the domain. Cui et al. [8] proposed a feature extraction technique for malicious URL detection using sigmoidal threshold and statistical analyses based on gradient learning. They showed the best accuracy of 98.7% with decision tree, SVM and Naïve Bayes classifiers. Zhao et al. [9] used Cost-Sensitive Online Active Learning(CSOAL) framework to minimize the imbalance of class in malicious URL detection.

Bet et al. [10] automated the classification of malicious URLs using the Naïve Bayes algorithm and presented various aspects of the URL classification process aimed at distinguishing between benign and malicious websites. In the course of their research, they provided detailed insights, highlighting that, in terms of accuracy, the Naïve Bayes method outperformed SVM.

Frank et al. [11] and Kapil et al. [12] used machine learning techniques and algorithms over large feature sets to address the problem at hand. Frank et al. [11] used a dataset with two million entries and three distinct feature sets based on whether the attributes of the URL are real-valued or binary. They used a range of machine learning algorithms, including RF, Multi-Layer Perceptron (MLP), C4.5, k Nearest Neighbor (kNN), SVM, C5.0, and Bayesian networks, across three distinct feature sets to tackle the problem of malicious URL detection. Their findings led to the conclusion that RF consistently outperformed the other algorithms regardless of the feature set employed. Notably, feature set 'A,' which comprises a combination of binary and real-valued attributes, exhibited the most favorable results across all classification techniques. Additionally, it was highlighted that MLP closely mirrored the performance of Random Forest.

Kapil et al. [12] used RF, J48, Bayesian Networks, and the Lazy algorithm classifier on a dataset consisting of 4,999 URLs categorized into five distinct URL classes, with a feature set comprised of 47 mixed attributes. Notably, their findings revealed that Random Forest and the Lazy algorithm exhibited close-knitted best results.

Apoorva et al. and Mamun et al. are noteworthy among researchers who have focused their studies on purely lexical features of the problem. [13] [14] Apoorva et al. [13] used RF, Gradient Boost, AdaBoost, Logistic Regression, Naïve Bayes with 21 different lexical features for binary classification with Random Forest performing the best with 90% accuracy. Mamun et al. [14] proposed a multi-classification of URL with 15 different lexical features They used kNN, J48, and RF classifiers. Their study was able to achieve the best accuracy of approximately 97% with RF.

In comparison to the related works, this paper makes an effort to detect malicious URLs purely based on their lexical features, which was not much addressed previously. We carefully selected three algorithms, namely Random Forest, XG Boost, and Light GBM, to assess how pure bagging classifiers compare to boosting classifiers in addressing the malicious URL detection problem. Also, we selected 23 different lexical features that focus on the semantic and lexical meaning of the URLs. Subsequently, we conducted an EDA to identify the most important lexical features that would effectively contribute to the predictions.

III. Methodology

The proposed methodology comprises four significant phases aimed at conducting a comprehensive analysis. First,

TABLE I
DISTRIBUTION OF URLS AMONG CLASSES IN THE DATASET

URL Type	Number of URLs	Percentage (%)
Benign	428,103	65.74
Defacement	96,457	14.81
Phishing	94,111	4.45
Malware	32,520	5

we initiate the process with the careful selection of a suitable dataset and the identification of appropriate classification techniques for our study. Next, we delve into the data through EDA and Data Visualization, which enables us to gain deeper insights and a better understanding of the dataset's characteristics. Following this, we move on to the critical step of Feature Selection and Classification, where we identify and prioritize relevant features and employ classification methods to construct predictive models that can classify URLs into different categories. Finally, in the Method Comparison phase, we assess and compare the performance of the various methods employed throughout the study, ultimately determining their effectiveness in achieving our objectives.

A. Selection of Dataset and Classification Techniques

The dataset utilized in this study is sourced from the Malicious URL dataset, as made available by [15]. This dataset comprises a total of $651,191$ URLs, each annotated with its corresponding class label. The distribution of URLs across different classes is detailed in Table I. This dataset serves as a critical foundation for our research, enabling an in-depth exploration of malicious URL lexical patterns and behaviors.

A comprehensive review of the existing literature on the classification of malicious URLs revealed a consistent trend wherein Random Forest consistently demonstrated higher performance across diverse scenarios within the problem domain. Notably, ensemble learning methodologies, particularly those rooted in boosting techniques, appeared to be underutilized in the context of malicious URL analysis. While some studies did incorporate conventional boosting methods like AdaBoost and Gradient Boost, there was a conspicuous lack of exploration into more advanced boosting techniques characterized by enhanced memory efficiency, speed, and overall performance, such as XGBoost and LightGBM, which emerged in the mid-previous decade. Recognizing the potential of these optimized boosting methods, we intentionally opted to investigate three distinct classification techniques: Random Forest, XGBoost, and LightGBM.This selection was motivated by our objective to discern whether the robust and optimized boosting methods, namely XGBoost and LightGBM, could outperform the well-established Random Forest algorithm. Furthermore, our study aimed to conduct a comparative assessment of both bagging and boosting methodologies in the context of the specific challenges posed by malicious URL classification.

TABLE II
LEXICAL FEATURES USED IN EDA

Lexical Feature	Data Type
Number of Digits in URL	Numeric
Google Index State (Whether URL is Google indexed or not)	Binary
Usage of URL Shortening Services	Binary
Occurrence of Suspicious Keywords (Win, Lottery, Paypal, payment, etc.)	Binary
IP Presence (Whether URL string Contains the IP address)	Binary
Number of Letters in URL	Numeric
First Directory Length	Numeric
Top Level Directory Length	Numeric
Number of Directories in URL	Numeric
Number of occurrences of ’WWW’	Numeric
Number of ’.’ In URL	Numeric
Number of ’/’	Numeric
Number of ’//’	Numeric
Number of ’@’	Numeric
Number of ’%’	Numeric
Number of ’?’	Numeric
Number of ’-’	Numeric
Number of ’_’	Numeric
Number of ’=’	Numeric
Overall Length of URL	Numeric
Occurrence of ’http’	Binary
Occurrence of ’https’	Binary
Hostname Length	Numeric

B. EDA and Data Visualization

In this stage, we undertook a two-phase approach. First, we conducted a comprehensive review of prior research to identify

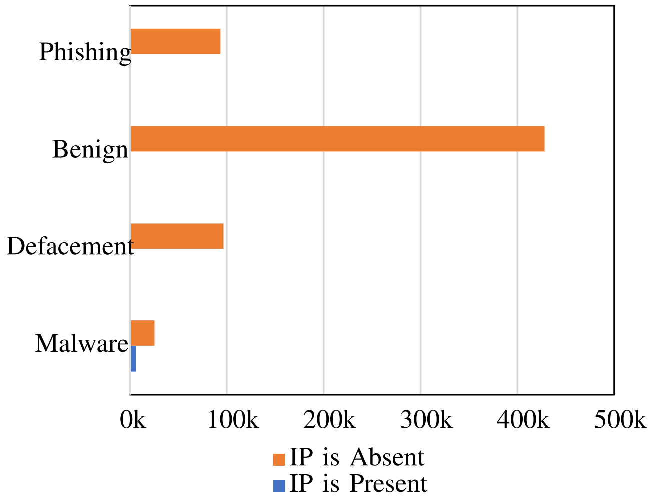

Fig. 2. IP Presence in URLs: X-axis denotes number of URLs with/without IP addresses; Y-axis represents URL types. Notably, only malware URLs include IP addresses

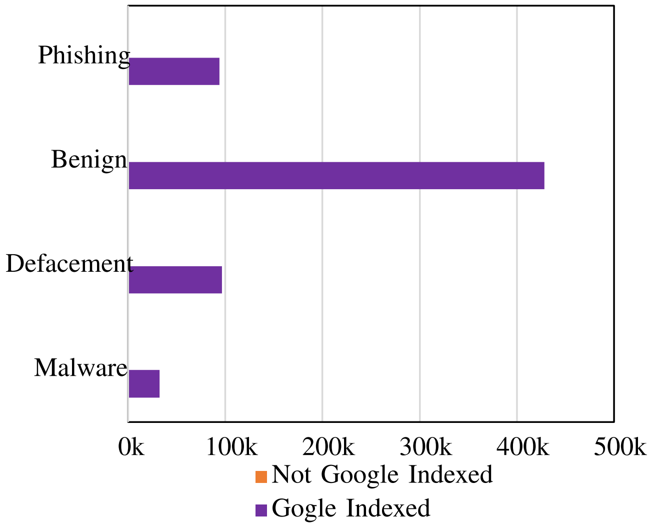

Fig. 3. Google Index Status: Y-axis denotes URL types, X-axis indicates Google-indexed count. Notably, all URL types are Google indexed

established lexical features commonly employed in similar studies. Second, we introduced some novel features based on our collective judgment, aiming to broaden the spectrum of lexical features explored in this research. Subsequently, we performed an in-depth EDA on this carefully selected set of 23 lexical features. To facilitate our analysis, we leveraged the data visualization capabilities of both Matplotlib and Seaborn. This investigation allowed us to gain a nuanced understanding of the distribution and characteristics of these lexical features across various classes of URLs. All the lexical features used to perform EDA are shown in Table II.

C. Feature Selection

Based on the results obtained by the EDA and the data visualization, we selected 21 features that would contribute to classifying URL as malicious type or benign.

Figure 2 depicts an example where certain malware instances exhibit a distinct characteristic of incorporating IP addresses within their URLs. This feature emerges as a noteworthy discriminator for classifying URLs as malware. Therefore, during the feature selection process, we deliberately retained features with similar discriminative attributes, incorporating them into our subsequent classification models.

In Figure 3, we present another data visualization underscoring a consistent pattern: all URLs, irrespective of their class, were indexed by Google. Consequently, features associated with such universal patterns, like the Google indexing status, were deemed non-informative and excluded from the final feature set. This strategic curation of features aimed to ensure a robust selection process, enhancing the resilience of our classification model.

D. Classification

Figure 4 presents the proposed malicious URL detection system. The detection phase contains two stages, which are training and detection respectively. After the EDA and feature extraction, the target encoding is performed to represent the class labels in a numerical way. The dataset is split into two parts as training data and testing data. The training set is used for training each classifier and, the model is then used for the classification of class for the URL in the test set.

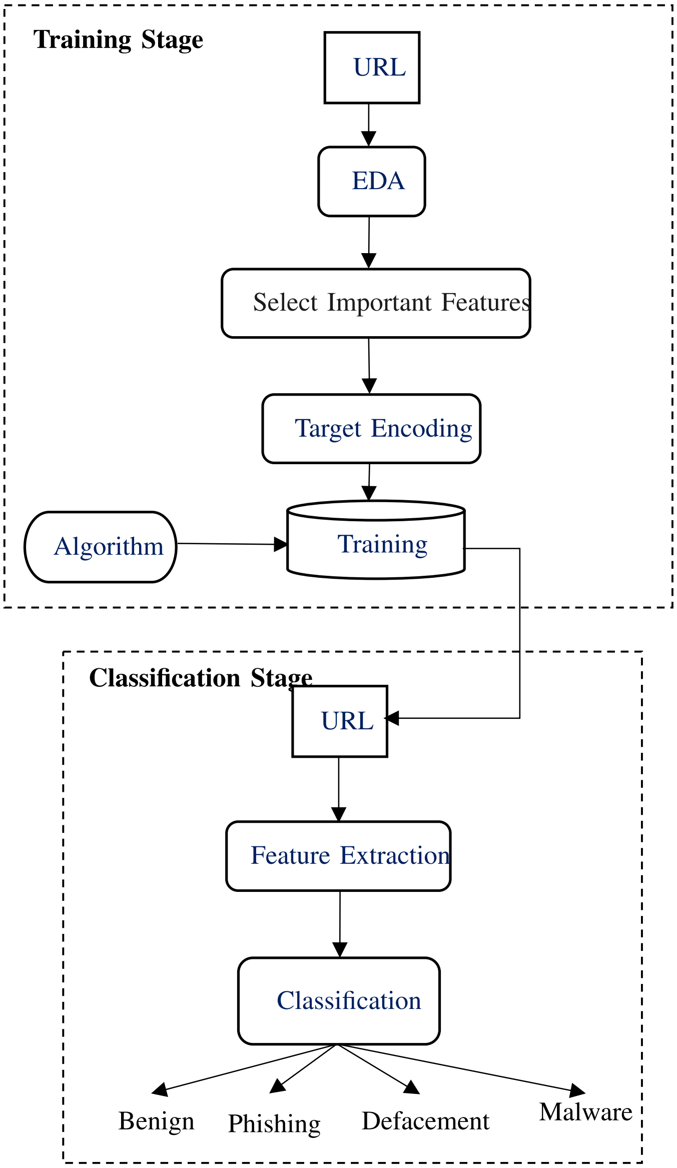

Fig. 4. Malicious URL Classification Design

In the classification stage, each input URL undergoes a feature extraction, and the extracted features are fed to the model to classify the class of the URL.

IV. Experimental Results

A. Experimental Setup

From the dataset of 651, 191 URLs, 80% of the data were used for training the model and the rest of the data were used to test the model. The shuffling and stratify methods were used to ensure the fair distribution of data from each class to the two separate data sets. Different parameters were used for each classifier.

B. Results and Discussion

In order to evaluate the models, four different metrics; accuracy, precision, recall and F1-score were used. Accuracy is the overall success rate of the method in terms of predictions (1). Precision is the ratio of positive predictions that are correctly classified (2). Low precision suggests the method is inclined to classify URLs as malicious, even when they are not. Recall is the ratio of actually positive cases that are also identified as such (3). F1-Score is the mean of precision and recall (4). High F1 value means the classifier is performing better.

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

$$\text{precision} = \frac{TP}{TP + FP} \tag{2}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{3}$$

$$\text{F1-Score} = \frac{2 * precision * recall}{precision + recall} \tag{4}$$

Equations 1 to 4 use the following definitions.

- TP : Number of true positives. Malicious URLs that are classified correctly.
- TN : Number of true negatives. Benign URLs that are classified correctly.
- FP : Number of false positives. Benign URLs misclassified as malicious.
- FN : number of false negatives. Malicious URLs that are misclassified as benign.

Low recall rates indicate that the concerned classification method is unable to detect malicious URLs. If low precision is detected in the results with high recall values present, it would imply that the method is classifying a high number of URLs as malicious. Hence, in the ideal scenario, both measures are expected to be high and also numerically closer for an unbiased and balanced prediction.

Table III shows the accuracy of each classification method while Table IV shows the precision, recall and F1 scores. Experimental results show that the Random Forest with 100 trees performs the best out of the three classification methods with 96.6% accuracy. Light GBM follows the performance of Random Forest with 95.6% accuracy, which is only 1% less than the best-performed classifier. XG Boost fails to come closer to the performance of the other two methods with only 93.2% accuracy. Random Forest being a bagging technique performs slightly better for the malicious URL detection and prediction problem compared to the two boosting techniques despite the latter's ability to achieve lower error rates and reduced biasness.

TABLE III
ACCURACY PER CLASSIFICATION METHOD

Classification Method	**Accuracy (%)**
Random Forest	96.6
Light GBM	95.6
XG Boost	93.2

Compared to the Random Forest and Light GBM, the XG Boost's precision and recall percentages are declined specifically for phishing and malware which are 76% and 73% respectively. For all the classification methods, hostname length is among the top five features that contributed to the prediction. Other common occurring lexical features are the number of directories and the first directory length. (Table V)

In comparison to the multi-classification method proposed by Mamun et al. that employed a set of 15 lexical features, which encompassed the application of kNN, C4.5, and Random Forest algorithms, their Random Forest model achieved an accuracy rate of approximately 97%. In our study, Random Forest model, which emerged as the best-performing method, attained an accuracy of 96.6%. Remarkably, our Random Forest method exhibited notably high precision and recall percentages across all malicious URL types, excluding malware, when compared to their results. A noteworthy observation is the absence of feature overlap between their selected lexical features and ours, highlighting the distinctive nature of our approach.

In comparison to the work conducted by Apoorva et al., our research represents a notable advancement. Their highest accuracy rate, achieved with the Random Forest algorithm, reached 92%, which is comparatively lower than the accuracy rate we have attained in our study. While they employed AdaBoost and Gradient Boosting techniques, achieving accuracy rates of 90% for both methods, we harnessed new and advanced boosting techniques; XG Boost and LightGBM, which yielded substantial improvements in accuracy, with rates of 93.2% and 95.6%, respectively. These findings highlight the progress made in our work, reinforcing the effectiveness of our proposed approach. Notably, Apoorva et al. relied on a set of 21 lexical features and did not employ an EDA process to filter out less significant features that may not contribute to the prediction of URL types. In contrast, our proposed approach adopted a more comprehensive feature selection process, ensuring that only the most relevant features were considered, thereby enhancing the accuracy of our classification models.

These findings highlight the notable progress achieved in our research, reinforcing the effectiveness of our proposed approach.

V. Conclusion

This paper presents a method for malicious URL detection using lexical features and machine learning techniques, incorporating prior EDA to build a robust model. The study explores a diverse set of lexical features and evaluates bagging and boosting techniques to enhance conventional malicious URL detection methods. The experimental results underscore

TABLE IV
PERFORMANCE OF EACH CLASSIFICATION METHOD BASED ON URL CLASS

URL Class	**Random Forest (%)**			**Light GBM (%)**			**XG Boost (%)**		
	Precision	***Recall***	***F1-Score***	***Precision***	***Recall***	***F1-Score***	***Precision***	***Recall***	***F1-Score***
Benign	97	98	98	97	99	97	95	98	97
Defacement	98	99	99	96	99	98	89	96	92
Phishing	99	94	97	96	89	92	92	76	83
Malware	91	86	88	90	85	85	88	73	80

TABLE V
TOP FEATURES CONTRIBUTING TO THE PREDICTION FOR EACH CLASSIFICATION METHOD

Random Forest		**Light GBM**		**XG Boost**	
Feature	***Percentage***	***Feature***	***Percentage***	***Feature***	***Percentage***
Number of occurrences of 'WWW'	12.67	First Directory Length	17.31	Top Level Directory Length	17.11
Hostname Length	12.57	Hostname Length	14.99	Occurrence of 'http'	12.72
Number of Directories in URL	12.46	Number of Digits in URL	11.78	Number of occurrences of 'WWW'	11.79
First Directory Length	7.59	Number of Directories in URL	11.45	Hostname Length	8.04
Occurrence of 'http'	7.25	Overall Length of URL	10.57	Number of Directories in URL	5.77

the effectiveness of Random Forest, achieving the highest accuracy, while Light GBM also performs impressively with only a 1% difference in accuracy compared to Random Forest. Notably, both classifiers exhibit higher precision and recall scores, ensuring well-balanced and unbiased prediction results.

Importantly, these impressive accuracy rates were obtained without the need for complex feature selection methods. Furthermore, the practical applications of this research extend to real-world scenarios, such as reinforcing user security in web browser extensions to ensure safer browsing experiences.

In our future experiments, we aim to assess classifier performance by varying parameter values from their defaults. We aim to determine if an optimal parameter configuration for Light GBM can surpass Random Forest's performance, which is the leading classifier for malicious URL detection. Additionally, we'll evaluate how Random Forest's performance is affected when altering the number of trees in the model, deviating from the default settings. This analysis will reveal the relationship between Random Forest's performance and the number of trees.

ACKNOWLEDGMENT

I wish to extend my gratitude to Dr. Udaya Wijenayake, my esteemed supervisor, whose role has been paramount in the successful completion of this paper. His multifaceted role as both my supervisor and co-author highlights the profound nature of his contribution. His unwavering commitment to guiding, mentoring, and actively collaborating throughout this study has been instrumental in shaping its outcome.

REFERENCES

[1] Symantec. (2019) Internet security threat report (istr) 2019. Accessed on [10/2023]. [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf

[2] S. C. H. Doyen Sahoo, Chenghao Liu, "Malicious url detection using machine learning: A survey," *School of Information Systems, Singapore Management University*, Jan. 2017.

[3] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2009.

[4] M. Al-Janabi, E. de Quincey, and P. Andras, "Using supervised machine learning algorithms to detect suspicious urls in online social networks," in *Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2017.

[5] I. N. V. D. Naveen, K. Manamohana, and R. Verma, "Detection of malicious urls using machine learning techniques," *International Journal of Innovative Technology and Exploring Engineering (IJITEE)*, vol. 8, no. 4S2, pp. 2278–3075, Mar. 2019.

[6] C. D. Xuan, H. Dinh, and T. Victor, "Malicious url detection based on machine learning," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 1, 2020.

[7] S. M. Nair, "Detecting malicious url using machine learning: A survey," *International Journal for Research in Applied Science and Engineering Technology*, vol. 8, no. 5, pp. 2670–2677, 2020.

[8] B. Cui, S. He, and X. Yao, "Malicious url detection with feature extraction based on machine learning," *International Journal of High-Performance Computing and Networking*, vol. 12, no. 2, 2019.

[9] P. Zhao and S. C. Hoi, "Cost-sensitive online active learning with application to malicious url detection," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2013, pp. 919–927.

[10] Sayamber, A. B., and A. M. Dixit, "Malicious url detection and identification," *International Journal of Computer Applications*, vol. 99, no. 17, pp. 17–23, 2014.

[11] F. Vanhoenshoven, G. Napoles, R. Falcon, K. Vanhoof, and M. Koppen, "Detecting malicious urls using machine learning techniques," in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2016.

[12] D. Kapil, A. Bansal, Anupriya, N. Mehra, and A. Joshi, "Machine learning based malicious url detection," *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 8, no. 4S, pp. 2249–8958, Apr. 2019.

[13] A. Joshi, L. Lloyd, P. Westin, and S. Seethapathy, "Using lexical features for malicious url detection - a machine learning approach," *Cryptography and Security*, Oct. 2019.

[14] M. S. Mamun, M. A. Rathore, A. H. Lashkari, N. Stakhanova, and A. A. Ghorbani, "Detecting malicious urls using lexical analysis," *Network and System Security*, pp. 467–482, 2016.

[15] M. Sidhdhartha. (2021) Malicious urls dataset. Accessed on 10/2023. [Online]. Available: https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset