Malicious Website Detection Using Random Forest and Pearson Correlation for Effective Feature Selection

Esha Sangra[1], Renuka Agrawal[2], Pravin Ramesh Gundalwar[3], Kanhaiya Sharma[4], Divyansh Bangri[5], Debadrita Nandi[6]
Department of Computer Science & Engineering, Symbiosis Institute of Technology,
Symbiosis International (Deemed University), Pune, India[1, 2, 4, 5, 6]
Department of Information Technology, Anurag University, Hyderabad India[3]

***Abstract*—In recent years, the internet has expanded rapidly, driving significant advancements in digitalization that have transformed day to day lives. Its growing influence on consumers and the economy has increased the risk of cyberattacks. Cybercriminals exploited network misconfigurations and security vulnerabilities during these transitions. Among countless cyberattacks, phishing remains the most common form of cybercrime. Phishing via malicious Uniform Resource Locator (URL)s threatens potential victims by posing as an imposter and stealing critical and sensitive data. An increase in cyberattacks using phishing needs immediate attention to find a scalable solution. Earlier techniques like blacklisting, signature matching, and regular expression method are insufficient because of the requirement to keep updating the rule engine or signature database regularly. Significant research has recently been conducted on using Machine Learning (ML) models to detect malicious URLs. In this study, the authors have provided a study highlighting the importance of significant feature selection for training ML models for detecting malicious URLs. Pearson correlation is employed in this study for selecting significant features, and the outcome demonstrates that in terms of accuracy and other performance indices, the Random Forest classifier outperforms the other classifiers.**

Keywords—Malicious URL; machine learning; feature selection; Random Forest, cybercrime

I. Introduction

Phishing is a form of social engineering when a cyber threat actor poses as a reliable individual or group in order to trick a user into disclosing private information or unintentionally allowing access to their network [1]. Some attack techniques that use malicious URLs include Drive-by Download, Phishing and Social Engineering, and Spam [2-4]. The potential outcomes include data breaches, loss of data or services, identity theft, malware infections, or ransomware attacks. Usually, blacklists have been the primary tool employed for such types of detection. [5] Nevertheless, blacklists cannot be considered comprehensive and cannot detect freshly generated malicious URLs. In recent years, there has been an increasing demand for evaluating machine learning methods to enhance the efficacy of malicious URL detectors [6]. Humans are the most common threat vector and are known to be the root cause of 74% of data breaches, according to Verizon's "2023 Data Breach Investigations Report [7]. An organization called APWG [8] that studies and disseminates information about malware and phishing scams, observed 1,077,501 phishing attacks in the last quarter of 2023. APWG recorded almost five million phishing attacks in 2023, which was deemed as the worst year for phishing activity. The Internet Crime Complaint Center [9] alone received a staggering approx. 300k reported phishing attempts. This number decreased from the previous year but increased significantly since 2018 when they received only 26,379 reports. Alameda Lost Nearly $200M to Phishing Attacks [10]. According to statistics presented, attacks using malicious URL techniques are ranked first among the ten most common attack techniques [11, 12]. URL phishing involves sending emails to redirect recipients to a fictious website and trick them into revealing sensitive data, such as confidential credentials or financial information to a malicious person. The website may appear legitimate, but its purpose is to exploit your trust by "phishing" for personal information that malicious actors can use for nefarious purposes. For example, an email containing a warning message of user activity on your bank account, credit card, or financial application. An email originating from an e-commerce or financial institution like Amazon or a bill desk warns about suspicious activity, such as a password breach. Users are redirected to click a URL to verify transactions or change their passwords. However, the link redirects them to a fake version of the application or website, where their login credentials are collected, or they are prompted to call "customer service". Phishing costs organizations millions to deal with malware and credential compromise situations and it also leads to productivity losses, further having a negative impact on company brand value. On an individual level there is financial loss and mental stress causing health complications. Phishing is considered the costliest attack. URL refers to resources on the Internet. In [13], Sahoo et al. URL is divided into protocol identifiers and resource names, which contain the IP address or the domain name pointing to the resource location.

A malicious URL is a variation of the original URL, which deceives the victim to visit the URL, leading to financial loss and theft of personal identification information such as identity, credit cards, etc. In recent years ML has played an important role in detecting malicious URL and overcoming some of the shortcomings of traditional methods. Large numbers of features degrade model performance in terms of latency, and the selection of features were not optimal, which leads in degrading the overall model accuracy.

The proposed study focuses on building models based on a set of appropriate features selected based on correlation, which will improve the overall trained model performance in terms of latency. This study determines whether the selecting subset of features has positive or negative impact on identifying malicious URLs with different machine learning algorithms. This is how remainder of the article is organized. Literature review is done in Section II. Next, in Section III, materials and methodology used for proposed model is discussed. A report on the experimental results obtained is covered in Section IV. The paper is concluded in Section V.

II. LITERATURE REVIEW

Many approaches have been developed in this area such as blacklisting, signature based, content-based classification, URL based classification. When machine learning is employed, the previous studies had different results for each algorithm and focused majorly on algorithm performance with all the extracted features. A tabular representation of work done by different researchers is shown in Table I.

TABLE I. LITERATURE REVIEW

Ref. No.	Technology	Dataset	Outcome and Limitation
[14]	Heuristics	16006 Benign and 5678 Malicious samples utilized	The increase in performance is accompanied by a false positive rate, which in practical settings generates a lot of false positive warnings. Besides this False negative rate was 46.15 % which was used for detection
[15]	List Based	5000 phishing websites from Phish Tank	NISOELM a unique method for phishing detection is proposed. Minor modification in the URL bypass the list and list must frequently be updated. To make sure that most malicious pages are identified with the presented information, the acquired knowledge must be updated on a regular basis as attackers modify their tactics.
[16]	Association rule based	collected over 1400 URLs from several sources and also 1200 phishing URLs from phishtank database	Large number of rules impact performance. Dataset consisted only of Binary attributes and only the phishing URLs are mined using the apriori algorithm for identifying the recurring patterns. No detection for newly arrived patterns.
[17]	Heuristics	Not specified	Proposed method consists of two processes, namely logo extraction and identity verification. Consider only 2 attribute website logo and domain name. Able to identify whether website logo and domain name are genuine or fake.
[18]	3 ML classifiers SVM, LR and Naïve Bayes	UCI machine learning repository, 11,055 URLs, each of 15uniquefeatures,	Performance impact due to model consider all feature for prediction. Naiye Bayes shows 100% accuracy but each feature weightage is same.
[19]	Machine learning	Not specified	Proposes a machine-learning framework for supporting intelligent web phishing detection and analysis, and provides its experimental evaluation. In particular we make use of state-of-the art decision tree algorithms for detecting whether a Web site is able to perform phishing activities. Performance impact due to model consider all feature for prediction
[20]	Machine learning	public dataset comprising 2.4 million URLs (instances) and 3.2 million features	Random Forest and Multi-Layer Perceptron attain the highest accuracy. Performance impact due to model consider all feature for prediction

III. MATERIAL AND METHODS

A URL consists of the protocol, subdomain, domain, path [21, 22]. Within the path, there can be filename, query param. Domain name can be broken down into domain and top-level domain. Malicious persons can add @ in the domain name or can use prefix/suffix in domain name. Length or depth of the URL is another feature which is exploited to deceive users. Use of HTTPS in the domain name to deceive the user into clicking the URL believing that it is a secure site. The structure of a URL is shown in Fig. 1.

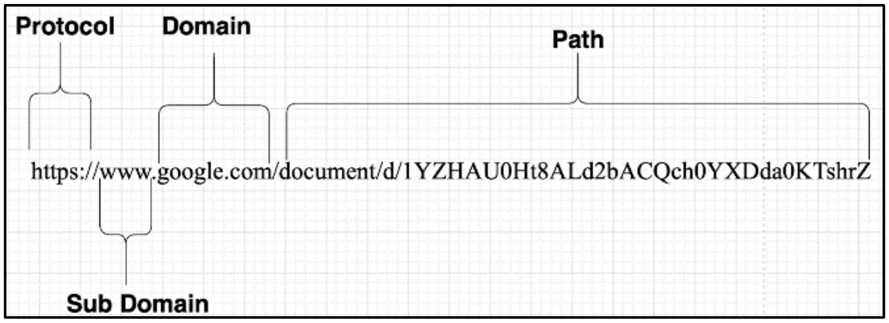

Protocol
Domain
Path
https://www.google.com/document/d/1YZHAU0Ht8ALd2bACQch0YXDda0KTshrZ
Sub Domain

Fig. 1. Structure of a URL.

Model for malicious URL detection is created using a jupyter notebook and serialized on disk in pickle format. Flask server was deployed to host the model and GET/POST route was defined to handle the incoming GET/POST request to render the UI for user input and user input is posted to the server for predicting the URL safe or not. Fig. 2 presents the malicious URL detection methodology steps. These steps are explained in next section. Also Fig. 3 represents the User interface created for detection of a URL to be malicious or not.

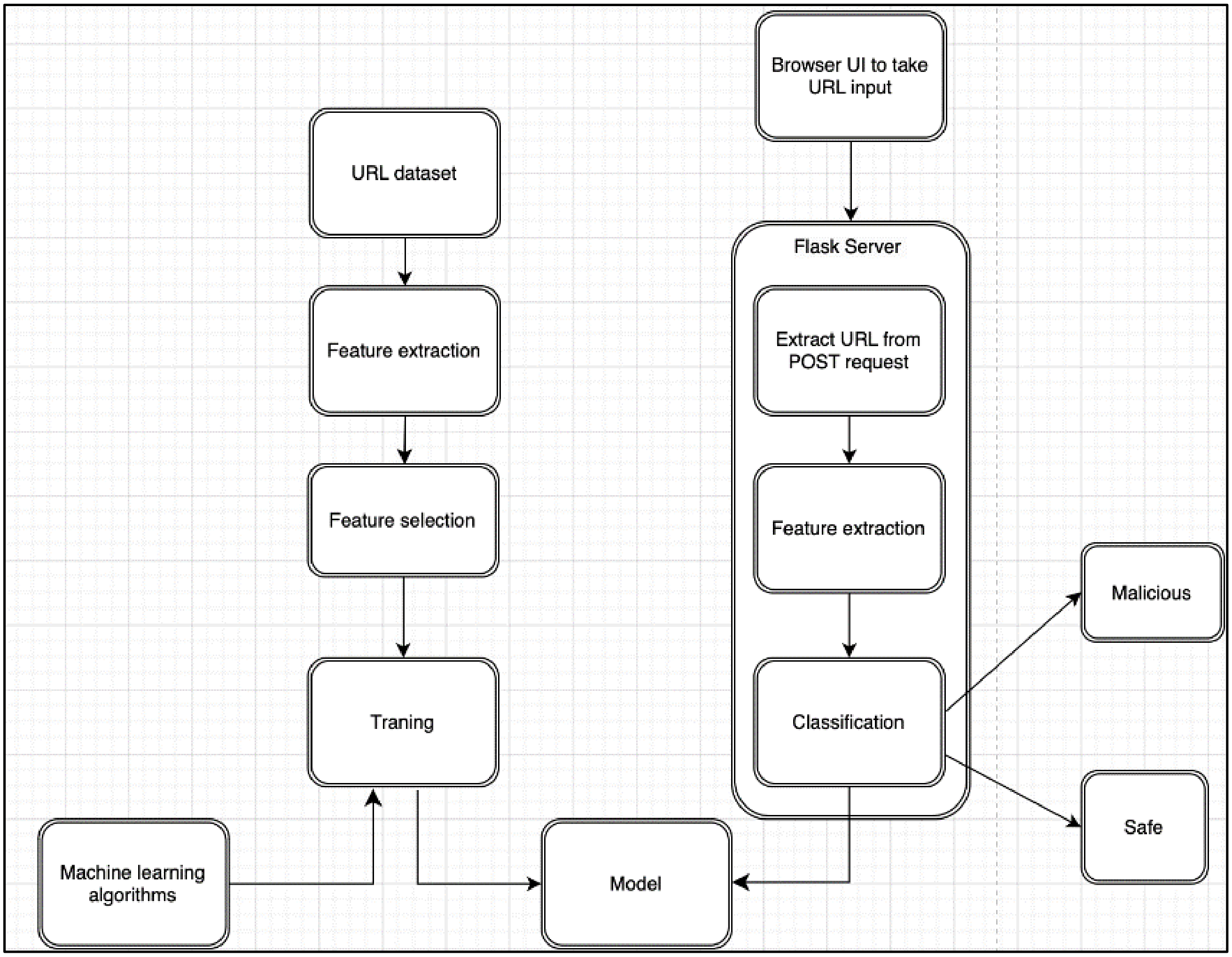
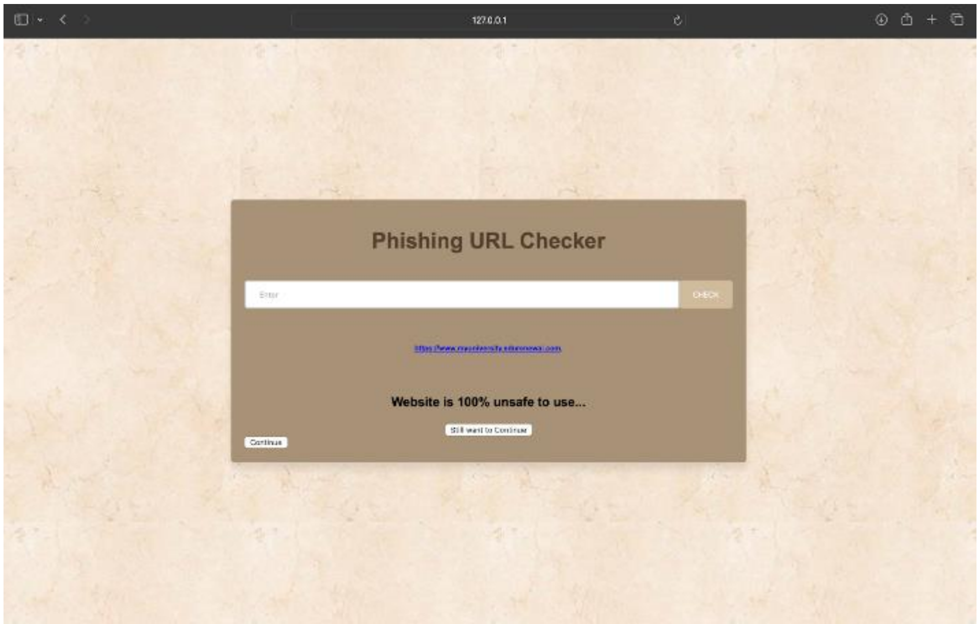


Fig. 2. Malicious URL detection flow.



https://www.[illegible].com

Website is 100% unsafe to use...

Fig. 3. User interface for URL checker.

A. *URL Dataset*

Suspicious URLs can be sent to Phishtank for verification https://www.phishtank.com/developer_info.php. The data in Phishtank is updated hourly. Phishtank is a free community site where anyone can submit, verify, track and share phishing data. This dataset is in the form of .csv file format. The models used in this manuscript dataset is fetched from phishtank. Besides this other source for similar dataset are available at https://urlhaus.abuse.ch/downloads/csv/ , URL has is a project from abuse.ch aiming at sharing malicious URLs being used for malicious software distribution, and dataset from Kaggle https://www.kaggle.com/datasets/sid321axn/malicious-urls-dataset, was used to host the data set for malicious and legitimate.

B. *Feature Extraction*

Feature extraction is one of the critical steps in the process of machine learning based malicious URL detection. Machine learning models require numeric value for training. For this purpose, essential characteristics of URLs are identified and passed to a function which converts the field value to 0 or 1 or other numeric values to distinguish malicious from benign. Tokenization and Lexical feature selection method is used for feature extraction Based on this criterion, the features are categorized in two different groups - Address bar based and domain-based features. In the case of Address bar-based features, features are selected from the lexical group of URLs, and are summarized in Table II. The address bar in web browsers is a powerful tool that goes beyond just entering website URLs. Below are the address bar features implemented in this project that goes beyond just entering website URLs. Tokenization is one of the techniques for feature extraction. It is defined as transforming a single string into a sequence of one or more non-empty substrings. Tokenization is performed utilizing the special characters (slash, dash and dot) in URLs. Once the token is extracted, it is passed to a function to check the characteristics such as DNS record validity or age of domain. This is characterized as Domain based features. Domain based features extracted from selected dataset is represented in Table III.

Selection of non-significant features can significantly impact a model performance besides increasing the model complexity. Selects a subset of relevant features while keeping the original feature space intact. The focus is on identifying the most informative features for modeling. The feature selection process is a step in building a machine learning model, performed by selecting a subset of the features in a set of extracted features. Feature selection aims to discover the most relevant and significant features for predicting the target variable. Feature selection has various benefits, such as Improved model interpretability, Reduced danger of overfitting, and improved model performance. Numerous methods for feature selection include filtering, wrapper approaches, and embedded approaches. Pearson correlation is employed and used to evaluate the model performance in the current work. Pearson correlation finds the correlation between features. Fig. 4 is the correlation matrix for the feature extracted to select the high-correlated and low-correlated features for training and evaluating the model.

TABLE II. DESCRIPTION OF ADDRESS BAR BASED FEATURES

S. No	Features	Description
1.	Domain	Extract the domain name
2.	Hostname/IpAddress	Parse the URL to extract an IP address . URLs may have IP address instead of domain name. Presence of an IP address alternative of the hostname name in the URL can be an indicator of malicious site
3.	@ symbol	In standard URL syntax, the "@" symbol is reserved for use in the format username@hostname. Anything before the "@" symbol is often interpreted as a username, and the browser ignores this part when resolving the URL Phishers exploit this behavior by inserting a legitimate domain name after the "@" symbol, making it appear as if the link leads to a trusted website. However, the actual website visited is determined by what follows the "@" symbol, not what precedes it.
4.	URL length	Phishers obscurate the URL by creating the long URL such that the user will not be able to differentiate a legit URL or malicious URL by masking the doubtful part of the address bar
5.	URL Depth	Computing the depth of a URL involves counting the number of levels or subdirectories in the URL path, typically separated by "/"
6.	Redirection //	"//" in a URL path reveals potential redirection or URL misconfiguration. Unexpected "//" positions could indicate unintended redirects or errors in URL formation.
7.	Http/Https	Phishers may add "HTTPS" to the domain (e.g., http://www.httpssecurelogin.com) to deceive users into believing a secure connection exists.
8.	URL Shortening Service	Services such as T2M, tine.be, Tiny URL, T.LT etc is a characteristic of malicious URL
9.	Prefix/Suffix ‘-’	Phishers use prefixes or suffixes to the domain name separated by some known separator such as "-" which makes it impossible for the user to distinguish that users feel that they are dealing with a legitimate website, for e: g www.example.com www.ex-ample.com

TABLE III. DESCRIPTION OF DOMAIN-BASED FEATURES

S. No	Domain features	Details
1.	DNS Record	WHOIS database does not recognize phishing websites identity or no records found for the hostname in DNS server
2.	Website Traffic	Top ranked websites are provided by Cisco Umbrella [26][27]. Alexa is no longer available. For the purpose of this research websites ranked among the top 100,000 is considered legitimate
3.	Age of Domain	Find the age of the domain by querying WHO database. Phishing websites are available for a short period. This research considers the minimum age of the legitimate domain, which is 12 months. Age here is nothing but different between creation and expiration time

The Pearson correlation coefficient, which is often denoted as r, is a measure of the linear correlation between two variables X and Y. It lies between -1 and +1. It is defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{l=1}^{n}(x_i-\bar{x})^2\sqrt{\sum_{i=1}^{n}(y_i-\bar{y})^2}}} \qquad (1)$$

Where

- n is sample size
- $x_i\ y_i$ are the individual sample points indexed with *i*
- $\bar{x} = \frac{1}{N}\sum_{i=1}^{n} x_i$ is the simple Mean for X
- $\bar{y} = \frac{1}{N}\sum_{1=1}^{n} y_i$ is the sample mean for Y

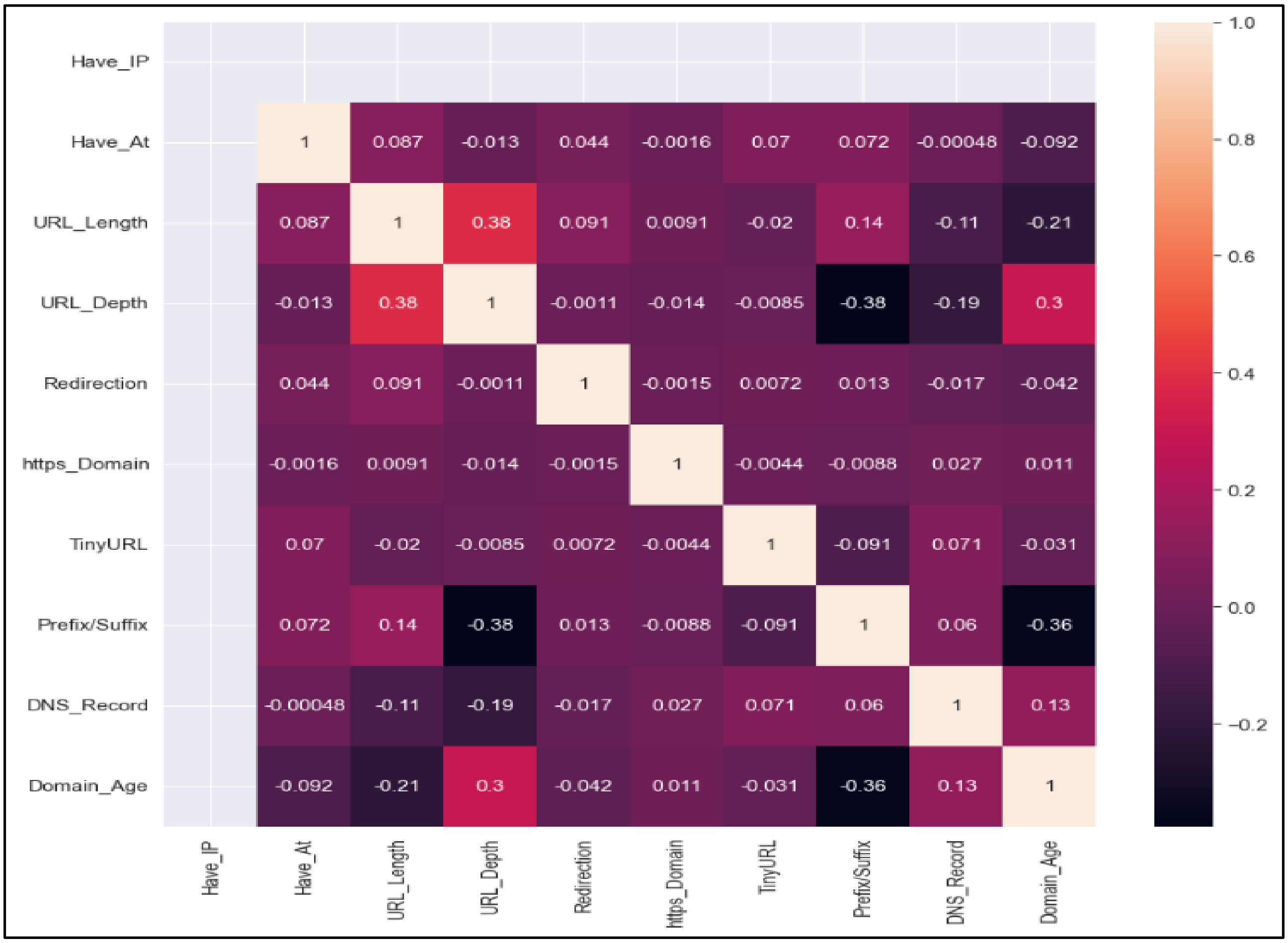


Fig. 4. Heat map of features.

For ease of modeling a threshold of .01 is chosen to filter significant features. Tabular representation of classification of features is shown in Table IV. Extracting the feature importance from the model which was created using all the features matches the correlated features. Table IV shows the segregation of features in URL based on the significance of the features. Graphical representation of importance of figures is shown in Fig. 5. The graph clearly depicts that URL features like *'URL_Depth'*, *'Domain_Age'* are most significant whereas features such as https_*Domain', Have_IP are* least significant in characterizing a URL as malicious.

TABLE IV. TABULAR REPRESENTATION OF CLASSIFICATION OF FEATURES

Feature set name	List of features
All features	{'Have_IP', 'Have_At', ' 'URL_Length', 'URL_Depth', 'Redirection', 'https_Domain', 'TinyURL', 'Prefix/Suffix', 'DNS_Record', 'Domain_Age', 'Domain'}
Top correlated Features	{'Domain_Age', 'DNS_Record' , 'Prefix/Suffix', 'URL_Depth', 'URL_Length }
Least correlated Features	{'https_Domain', 'Redirection', 'Have_IP', 'TinyURL', 'Have_At'}

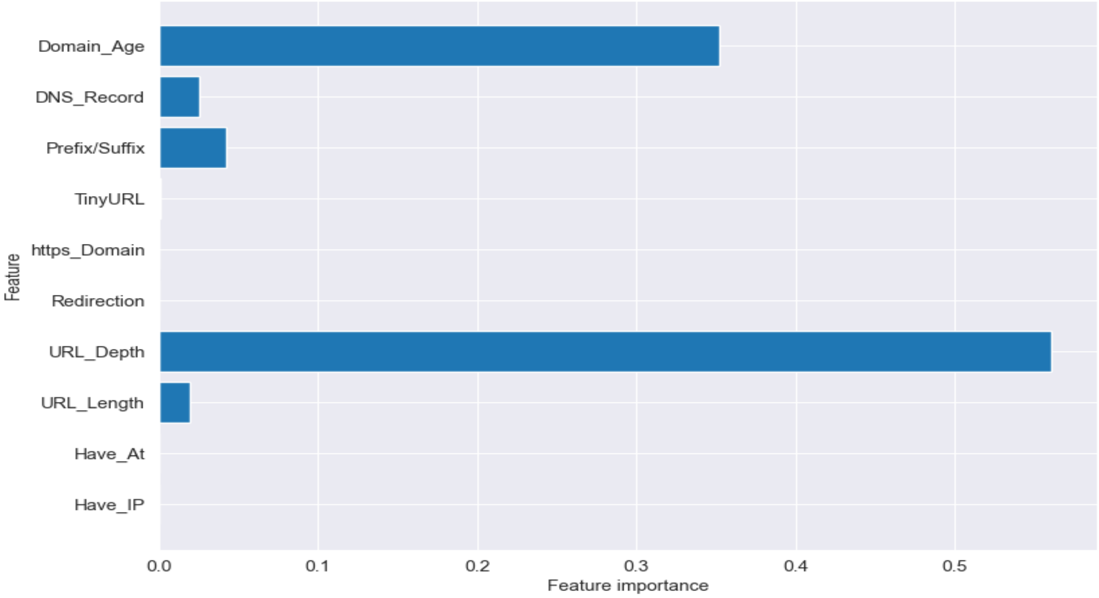


Fig. 5. Importance of feature.

C. Model Development and Performance Evaluation

Machine learning algorithms for detecting malicious URLs have been studied and are widely applied [23,24]. Supervised machine learning algorithms are classified and regression. This data set comes under classification problems, where the input URL is either phishing 1 or legitimate 0. The supervised machine learning models considered for training the dataset in this notebook are, Decision Tree and Random Forest. The model was trained with a decision tree and random forest algorithm with “all features”, “Top correlated features”, and “Least correlated features”, as outlined in Table IV earlier. Decision trees are widely employed models for classification and regression-related tasks. Fundamentally, they learn a hierarchy of if/else questions to determine a decision. Learning a decision tree implies learning the pattern of if/else conditions that optimally lead to the true answer. In the machine learning setting, these questions are called tests (not to be confused with the test set, which is the data that is used to test to interpret the model generalizability. A decision tree consists of nodes representing decisions on features, branches representing the result of these decisions, and leaf nodes representing predictions. Internal nodes are an examination of a feature, and each branch corresponds to the outcome of the test, and each leaf node fits a class label. A random forest which is an ensemble model of decision tree, works by creating multiple decision trees. The idea behind random forests is to build a tree using random samples from the training dataset. The random forest combines the output of individual decision trees to generate the final output by averaging their results. They are powerful, often work well without heavy tuning of the parameters, and don’t require data scaling. The entire data set of URLs containing legitimate and phishing URLs is then divided into 4 variables, X_train, X_test, Y_train, Y_test using the ‘sklearn.model_selection’ module/library. X_train: This variable holds the features (input variables) for the training set. In this paper, use these features to train your machine learning model. X_test: This variable holds the features for the testing set. In this paper, these features evaluate the performance of your trained model on unseen data. y_train: This variable holds the target variable (output variable) corresponding to the training set. It contains the expected outcomes for the training data. y_test: This variable holds the target variable corresponding to the testing set. It contains the expected outcomes for the testing data, which you use to compare against the predictions made by your trained model. Following model performance metrics are captured for performance assessment. True Positive (TP), False Positive (FP), True Negative (TN) and False Negative(FN) are some of the variables defined in confusion matrix. These are used for calculating the performance of a machine learning classification model as are used in Eq. (2)-Eq. (5) [28]. In context to calculating the ML model performance for detection of URL as malicious or genuine the performance measures are defined as

1) *Precision:* It is the ratio of true positive URL among the total number of positive URL predicted

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

2) *Recall*: It is the ratio of predicted true URLs and the total number of actual true URL which is sum of true positive and false negative predicted URL.

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

3) *F1 Score*: It is the harmonic mean of precision and recall.

$$\text{F1 Score} = 2.\frac{Precision.Recall}{Precision+Recall} \tag{4}$$

4) *Accuracy*: Success rate of the URL prediction technique and is coined as the ratio of True predicted to all the the samples in the dataset

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (5)$$

IV. Model Setup and Results Obtained

Setup environment: Operating System - Mac OS, Language - Python 3.12.2 Web framework - Flask, Model builder = Jupyter notebook, ML framework/tools - Pandas, scikit-learn, Numpy Hardware: RAM 16 GB 3733 MHz LPDDR4X; 2 GHz Quad-Core Intel Core i5. The experiment extracted the features from the legitimate URL and phishing URL data set and labeled accordingly. The data set used for the experiment is of 10k records which include the 5k phishing and 5k legitimate URL. Fig. 6 shows the distribution of the individual features values in the dataset used for detection of malicious URL [25].

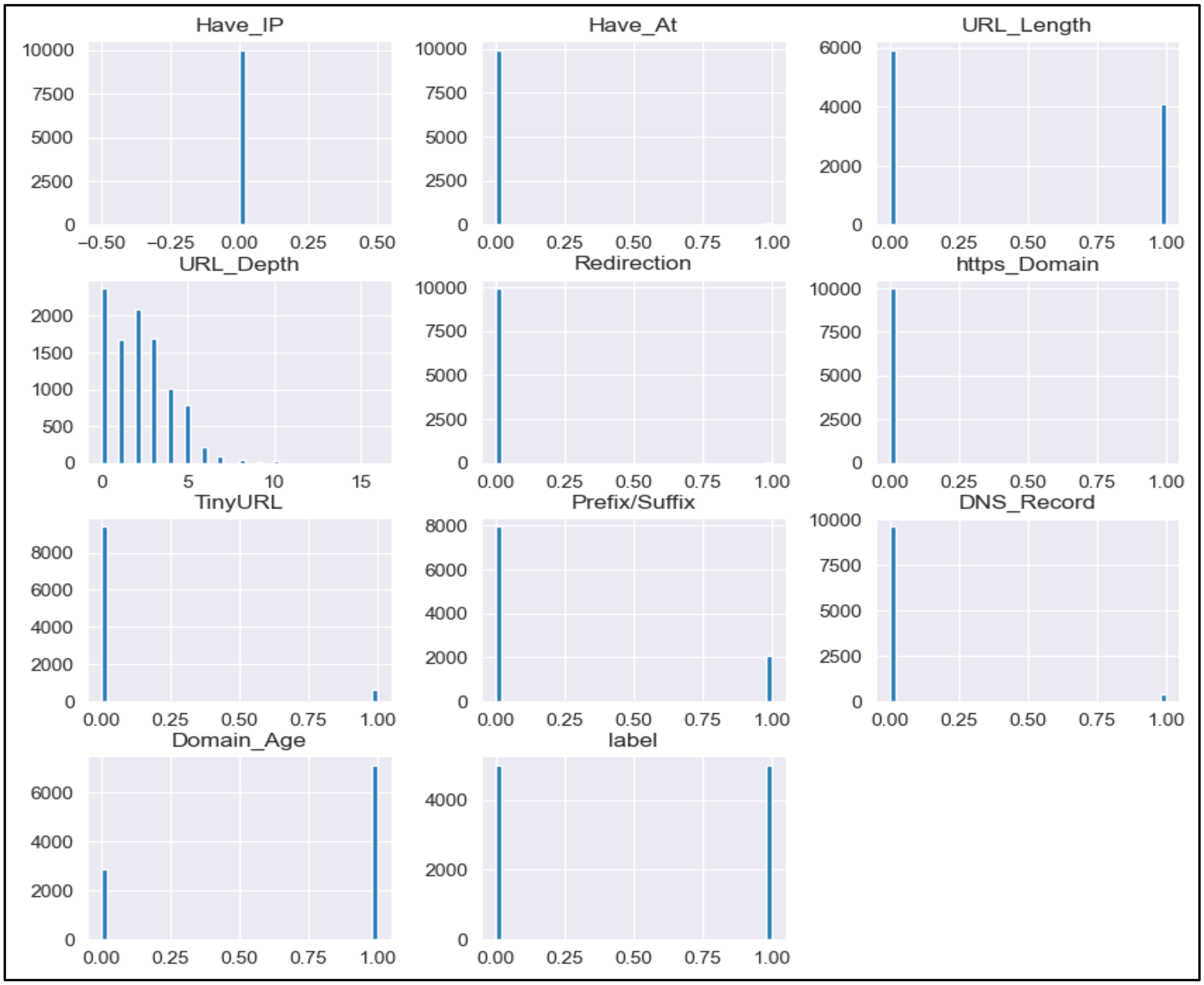


Fig. 6. Feature distribution.

In this paper, models were trained using machine learning models of decision tree and Random Forest after splitting dataset into training and testing with 80% of data used for training the model and tested on 20% of data with following set of features

- ALL
- TOP correlated
- LEAST correlated

Table V and Table VI shows the performance parameters obtained from Decision tree and its ensemble version Random Forest model. The parameters considered for assessment of models are accuracy, F1 score, Recall and Precision are best for the decision tree. Below is the metric for performance assessment of Decision Tree model based considering ALL, TOP and LEAST correlated features. Tabular representation of performance parameters discussed earlier for Decision tree ML model is represented in Table V.

TABLE V. Decision Tree Model

	ALL	TOP	LEAST
Accuracy	0.913	0.899	0.504
F1 Score	0.908	0.884	0.035
Recall	0.853	0.820	0.018
Precision	0.970	0.959	0.934

Because latency has a direct impact on how well systems operate in real time, it is also considered as a significant parameter for selecting one model over another [26]. Lower latency is preferable. The wait time for a result is known as latency. A ML model is not considered good if there is a noticeable waiting period before the occurrence of the responses. Improving latency is crucial since every system aspires to operate in real time [27]. An analysis of the time taken (latency) by the model to test 20% of the data where Decision Tree model is built using different set of features is as below.

- All features = 0.0033ms
- Top correlated = 0.0016ms
- Least correlated = 0.0026ms

Considering an ensemble model of Decision tree which is Random forest, performance parameters are rechecked.

Tabular representation of results obtained using ensemble model is represented in Table VI.

TABLE VI. RANDOM FOREST MODEL

	ALL	TOP	LEAST
Accuracy	0.953	0.947	0.504
F1 Score	0.946	0.924	0.035
Recall	0.876	0.873	0.018
Precision	0.978	0.967	0.857

Table VI shows the performance of Random Forest model for detection of URL as malicious or not. Latency by the Random forest model to test 20% of the data where Random Forest model is built using different set of features are obtained as follows:

- All features = 0.0027ms
- Top correlated = 0.0010ms
- Least correlated = 0.0020ms

TABLE VII. COMPARATIVE STUDY WITH EXISTING RESEARCH WORK DONE

Reference No.	Multiple ML models Used	Results:						
		Accuracy	F1 Score	Recall	Precision	Features in Modelling	Model Complexity	Latency Considered
[28]	yes	NA	High	High	High	Considered all features	High	No
[29]	yes	High	High	High	High	Not considered	NA	No
[30]	Yes	High	NA	NA	Good	Lexical features	NA	No
[31]	Yes	High	High	High	High	semantic and contextual features	High	No
Proposed work	Yes	High	High	High	High	Considered only significant Features without compromising on Performance	Reduced as only significant Features considered.	Yes

Decision tree and Random Forest model metric are similar and also perform similarly with the given URL dataset for selected feature sets. Random Forest uses a default estimator=100 of trees on a URL dataset. Ensemble model of Decision tree which is Random Forest performance is better in terms of performance indices as well as in terms of computation time. Besides this by reducing the number of features it can be clearly stated that the performance of model remains unaffected by reducing the number of features and selection only significant features for models designing. This will also reduce model complexity without compromising on model performance. However, the using the top correlated features shows significant model performance improvement in both Decision Tree and Random Forest. Table VII shows a comparative study on the model proposed and those used by researchers in similar domain.

V. CONCLUSION

The early systems were dependent upon patterns of known malicious URLs, rule-based methods. These systems are excellent in protecting the user from known malicious URLs but are inefficient in securing them from new emerging attacks. Although some attempts were made to build a model using ML but due to resource intensive, there is inefficiency in ML based malicious URL detection because the models have mostly considered either all the features which including non-correlated features or least significant features as well. While accessing performance the models performs well but fail to justify the response time or latency of the model. In this study, a comprehensive study on URLs like phishing or legitimate is used to analyze ML models based on the different feature selection and a study on the impact of feature selection is done. By using all the features or using only the most correlated features have slight impact on the performance of model parameters accuracy, F1 score, recall and precision but the difference in the model latency is quite significant with most correlated features and all features. This shows that using all the features impact the URL detection performance significantly with minimal gain in accuracy. Using highly correlated features helps in reducing the number of features which leads to reduction in model complexity and will further improve the model performance in terms of latency with minimal or negligible impact on the model performance.

REFERENCES

[1] Breda, Filipe & Barbosa, Hugo & Morais, Telmo. (2017). SOCIAL ENGINEERING AND CYBER SECURITY. 4204-4211. DOI:10.21125/inted.2017.1008.

[2] M. Khonji, Y. Iraqi, and A. Jones, “Phishing detection: a literature survey,” IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013. [3] 10.1109/SURV.2013.032213.00009

[3] M. Cova, C. Kruegel, and G. Vigna, “Detection and analysis of driveby-download attacks and malicious javascript code,” in Proceedings of the

19th international conference on World wide web. ACM, 2010, pp. 281–290. https://doi.org/10.1145/1772690.1772724.

[4] R. Heartfield and G. Loukas, "A taxonomy of attacks and a survey of defence mechanisms for semantic social engineering attacks," ACM Computing Surveys (CSUR), vol. 48, no. 3, p. 37, 2015. https://doi.org/10.1145/2835375

[5] Hameed, W & Ahmed, I & Khan, B & Kumar, Raja. (2017). USING BLACK-LIST AND WHITE-LIST TECHNIQUE TO DETECT MALICIOUS URLS. 10.26562/IJIRIS.2017.DCIS10081.

[6] Oshingbesan, Adebayo & Okobi, Chukwemeka & Ekoh, Courage & Richard, Kagame & Munezero, Aime. (2021). Detection of Malicious Websites Using Machine Learning Techniques. 10.13140/RG.2.2.30165.14565.

[7] C. David Hylender, Philippe Langlois, Alex Pinto, Suzanne Widup, "Data Breach Investigation report by Verizon Business" ,2024.

[8] APWG Phishing Activity Trends Report, Phishing Activity Trends Report, 4th Quarter 2023, Unifying the Global Response To Cybercrime apwg_trends_report_q4_2023

[9] Internet Crime Report 2023 by FEDERAL BUREAU OF INVESTIGATION. Internet crime complaint center. 2023_IC3Report.pdf

[10] H. M. Junaid Khan, Q. Niyaz, V. K. Devabhaktuni, S. Guo and U. Shaikh, "Identifying Generic Features for Malicious URL Detection System," *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York, NY, USA, 2019, pp. 0347-0352, doi: 10.1109/UEMCON47517.2019.8992930.

[11] Cho, Do & Dinh, Hoa & Victor, Tisenko. (2020). Malicious URL Detection based on Machine Learning. International Journal of Advanced Computer Science and Applications. 11. 10.14569/IJACSA.2020.0110119.

[12] M. Aljabri *et al*., "Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions," in *IEEE Access*, vol. 10, pp. 121395-121417, 2022, doi: 10.1109/ACCESS.2022.3222307

[13] D. Sahoo, C. Liu, S.C.H. Hoi, "Malicious URL Detection using Machine Learning: A Survey". https://doi.org/10.48550/arXiv.1701.07179

[14] Seifert, Christian & Komisarczuk, Peter & Welch, Ian. (2009). Identification of Malicious Web Pages with Static Heuristics. 10.1109/ATNAC.2008.4783302.

[15] Yang, Liqun & Zhang, Jiawei & Wang, Xiaozhe & Li, Zhi & Li, Zhoujun & He, Yueying. (2020). An improved ELM-based and data preprocessing integrated approach for phishing detection considering comprehensive features. Expert Systems with Applications. 165. 113863. 10.1016/j.eswa.2020.113863.

[16] Jeeva, Carolin & Rajsingh, Elijah. (2016). Intelligent phishing url detection using association rule mining. Human-centric Computing and Information Sciences. 6. 10.1186/s13673-016-0064-3.

[17] Chiew, Kang Leng & Chang, Ee & Sze, San & Tiong, Wei. (2015). Available online Utilisation of website logo for phishing detection. Computers & Security. 54. 10.1016/j.cose.2015.07.006.

[18] Wejinya, Gold & Bhatia, Sajal. (2021). Machine Learning for Malicious URL Detection. 10.1007/978-981-15-8289-9_45

[19] Cuzzocrea, Alfredo & Martinelli, Fabio & Mercaldo, Francesco. (2019). A machine-learning framework for supporting intelligent web-phishing detection and analysis. IDEAS '19: Proceedings of the 23rd International Database Applications & Engineering Symposium. 1-3. 10.1145/3331076.3331087.

[20] Nana, S.R., Bassolé, D., Dimitri Ouattara, J.S., Sié, O. (2024). Characterization of Malicious URLs Using Machine Learning and Feature Engineering. Social Informatics and Telecommunications Engineering, vol 541. Springer, Cham. https://doi.org/10.1007/978-3-031-51849-2_

[21] Hawkins, John., 4th International Conference on NLP Trends & Technologies (NLPTT 2023) - Data Science & Cloud Computing Track (DSCC)At: Chennai, India

[22] Vrbančič, Grega & Fister jr, Iztok & Podgorelec, Vili. (2020). Datasets for phishing websites detection. Data in Brief. https://doi.org/10.1016/j.dib.2020.10643833.

[23] Cui, Baojiang & He, Shanshan & Yao, Xi & Shi, Peilin. (2018). Malicious URL detection with feature extraction based on machine learning. International Journal of High Performance Computing and Networking. 12. 166. 10.1504/IJHPCN.2018.094367.

[24] Shantanu, B. Janet and R. Joshua Arul Kumar, "Malicious URL Detection: A Comparative Study," 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 2021, pp. 1147-1151, doi: 10.1109/ICAIS50930.2021.9396014.

[25] Cho Do Xuan, Hoa Dinh Nguyen and Tisenko Victor Nikolaevich, "Malicious URL Detection based on Machine Learning" International Journal of Advanced Computer Science and Applications, 11(1), 2020. http://dx.doi.org/10.14569/IJACSA.2020.0110119.

[26] Külzer, D.F., Debbichi, F., Stańczak, S. and Botsov, M., 2021, June. On latency prediction with deep learning and passive probing at high mobility. In *ICC 2021-IEEE International Conference on Communications* (pp. 1-7). IEEE. 10.1109/ICC42927.2021.9500495

[27] Bezerra, D., de Oliveira Filho, A.T., Rodrigues, I.R., Dantas, M., Barbosa, G., Souza, R., Kelner, J. and Sadok, D., 2022. A machine learning-based optimization for end-to-end latency in TSN networks. *Computer Communications*, *195*, pp.424-440. https://doi.org/10.1016/j.comcom.2022.09.011.

[28] Reyes-Dorta, N., Caballero-Gil, P. & Rosa-Remedios, C. Detection of malicious URLs using machine learning. *Wireless Netw* (2024). https://doi.org/10.1007/s11276-024-03700-w

[29] Vundavalli, V., Barsha, F., Masum, M., Shahriar, H. and Haddad, H., 2020, November. Malicious URL detection using supervised machine learning techniques. In *13th International Conference on Security of Information and Networks* pp1-6. https://doi.org/10.1145/3433174.3433592

[30] A. Saleem Raja, R. Vinodini, A. Kavitha,Lexical features based malicious URL detection using machine learning techniques, Materials Today: Proceedings,Volume 47, Part 1,2021, Pages 163-166,ISSN 2214-7853, https://doi.org/10.1016/j.matpr.2021.04.041.

[31] Lixiao Jin, Ruiyang Huang, Xuanming Zhang, Fangjie Wan," A Malicious URL Detection Method Based on Bert-CNN", Advances in Transdisciplinary Engineering. Electronic Engineering and Informatics , Vol 51, page no. 515-522, doi 10.3233/ATDE240115