[1] **Nachiyappan S**

[2] **Pradeep K V**

[3] **Rajarajeshwari S**

[4] **Sujay Doshi**

[5] **D. Yuvaraj**

Ensemble Learning-Based Browser Extension for Mitigating Cyber Attacks Carried out using Malicious Short URLs

Abstract: - As digitization continues to expand and cybercrimes become more prevalent, making it is crucial to prioritize the implementation of robust security measures. Malicious short URLs are frequently utilized as a vector for cyber-attacks on online forums and social media platforms. To address this issue, a plugin-based solution that uses ensemble learning to combine random forest, k-neighbors classifier, and logistic regression into a stacked model, was developed. The model was trained over the combination of three most popular kaggle datasets, with over 1081195 URLs. Additionally, gradient boosting was applied to further enhance the model's performance, resulting in a 92% accuracy in the detection. We developed the browser extension with Flask and JavaScript that identifies URLs as malicious or safe, for facilitation of the proposed solution. The work emphasizes the need for effective measures to mitigate cyber-attack risks, particularly those involving malicious short URLs.

Keywords: Cybersecurity, Short URLs, Malicious URL detection, Feature extraction, Ensemble learning.

I.INTRODUCTION

Recent studies on cyber threat defence show that prevalence of cybercrimes is steadily increasing. These statistics showed that 80.7% of all computer systems had at least one security breach by the year 2020 [1]. With the advent of modern technology and interconnectedness of devices cyber threats are escalating day by day. Exploitation of the formed vulnerabilities grants the cyber criminals access to sensitive information.

Increase in social engineering attacks like phishing and spear-phishing has also been observed in the recent years, which has led to financial losses for both individuals and industries alike [2]. As we can see in Figure 1, the trend displays an increase in the frequency of phishing attacks per month. Phishing attacks totaled 1,270,883 in the third quarter of 2022, setting a new record and making it the worst quarter ever recorded [4]. A phishing message was the first step in 81 percent of malware attacks of corporate equipment [3].

[1] *Corresponding author: School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India. Email: nachiyappan.s@vit.ac.in

[2] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India. Email: pradeep.kv@vit.ac.in

[3] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India.

Email: rajarajeswari.s@vit.ac.in

[4] School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, Tamil Nadu, India.

Email: sujay.doshi2023@vitstudent.ac.in

[5] Department of Computer Science, Cihan University-Duhok, Duhok, Iraq. Email: d.yuvaraj@duhokcihan.edu.krd

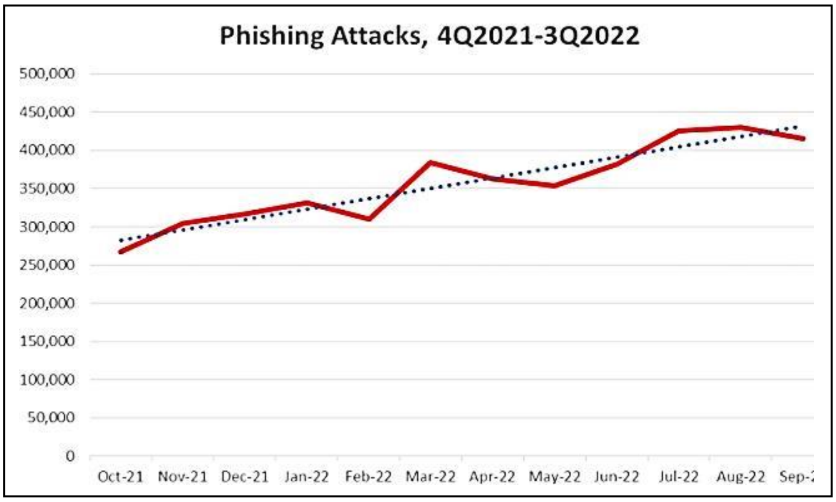
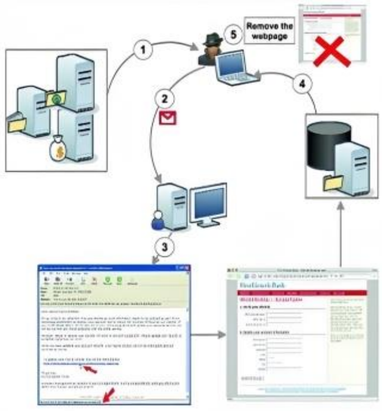


Figure. 1. Phishing Trend Report [4]

Phishing is a fraudulent activity that combines technical deception and social engineering. The main goal of the attack is to steal the user's personally identifiable information by mimicking of legitimate websites and creating an environment where the user believe it is required to enter the information requested.



Phishing life cycle: (1) phisher copies the content from legitimate site and constructs the phishing site; (2) phisher sent link of phishing URL to Internet user; (3) user opens the link and fills personal on fake site; (4) phisher steals the personal information of user; (5) phisher deletes the fake web page

Figure. 2. Lifecycle of Phishing Attack

With increase in sophisticated modalities for obtaining sensitive information, in recent years it has been observed that URL shortening methods have been used to mask malicious URLs. Scammers are masquerading the actual URLs, i.e. the user doesn't know the actual link behind the short URL [5]. Detection of such masked URLs is critical as well as challenging since the URLs length is brought down by as much as 91% in length and hence the number of features extracted are very limited [6].

URL shortening services were introduced in 2001, and its wide popularity has led to development of more than 500 shortening services, the prominent ones being Bitly and TinyURL. The initial purpose of URL shorteners was to prevent the breakage of long, complex URLs while copying text, and to make it easier to share content with long URLs. Cyber criminals leverage these shorteners to obfuscate the actual malicious URL.

In this research, we propose an approach to classify long as well as short URLs. To further facilitate this use of the solution, we plan to deploy it in the form of a web-based plugin, to detect real time malicious URLs. Usage of an ensemble learning model was done to achieve 92% accuracy alongside a low false positive rate on the dataset gathered. The browser will detect the URL being malicious or benign and highlight its detected response in red or green colors. Added functionality in the plugin also not allows the user to click the link by mistake and be exposed to the threat. The proposed methodology also uses the local cache for faster and better classification and interaction with the user. The approach can be used on any Chromium-based browser for real time detection and mitigation.

The remainder of the paper is organized as follows; Section II presents the Existing Architecture along with its flaws. Section III presents the Proposed Methodology and Implementation. Section IV presents the Results and Discussion related to the study, while Section V presents its conclusion. Section VI presents the future scope and improvements that could be done.

II.EXISTING ARCHITECTURE

2.1 Related Works

Rima Masri, et. al.[7] in their paper, propose and implement a system for automatically detecting malicious advertisements employing three different online mal- ware detection systems i.e., VirusTotal, URLVoid and TrendMicro. In this paper, advertisements are collected using Selenium-based web crawler which is a web browser automation tool that allows a programmable control over the web browser. Each URL crawled is scanned by these tools which use blacklisting re-positories for detecting malicious URLs. This study aims to detect malicious online ads using different detection and analysis tools. URLVoid tools boast the best accuracy of 73% with the highest true positive value, and the lowest false negative value. Malicious ads are not removed or blocked. It is unable to detect newly generated malicious URLs and is ineffective against zero-day exploit as blacklisting repositories are not updated. This system does not detect shortened URLs until that URL is present in the blacklisted repository.

Manan et. al.[8] in their paper, characterize features of URLs based on URL itself, content, network protocol and HTTP Response from the URL. URL-based features target the lexical and host-based features while the content-based one targets HTML, JavaScript features. The system proposed several methods to detect malicious URLs or malicious JavaScript in webpages. This paper implements feature extraction on URLs that have been extracted from web. This method replaces the black listing repository method (as they are vulnerable to newly generate malicious URLs and zero-day exploit). It also targets HTML, JavaScript and Network Protocol based features for detecting malicious URLs. Websites with special or unicode characters are not taken into consideration. No methods to detect malicious shortened URLs. Extracted features are not trained using ML models for better results. This paper used feature extraction to train their model which helped us in our feature extraction.

D. Liu et. al.[9] in their study, use a CNN (Convolutional Neural Network) to learn and recognize images of malicious webpages. This method adopts the perspective of uses and takes screenshots of malicious webpages to invalidate web spams. The model has an input layer where screenshots are placed and three convolution layers are activated using Rectified Linear Unit (ReLU) and maximally pooled to prevent overfitting. This model is better than classic ML models like Random Forest and SVM (which cannot detect redirection spam and hidden IFrame spam which leads to low detection accuracy). The proposed method is cost efficient and very fast. CNN model reduces false positive rate and increases cover- age rate. The CNN model has a precision of 91.84% and F1 of 92.71%. Malicious URLs have to be visited to take screenshots of websites for training the model, thus might harm the system. Changing the design of website is enough to bypass the model and this model is computationally intensive as it uses images instead of text.

Pttewar et. al.[10] discuss about short URLs and ways to detect malicious short URLs and shortening services like Bitly, TinyURL, Link and goo.gl. The author discusses about approaches to detect malicious URLs using blacklist, heuristic approaches and using machine learning which implements Feature Selection on short URLs using blacklist, lexical, host-based and content features even including HTML, JavaScript and Visual features. Short URLs are also classified as benign or malicious using classifiers and click traffic data. The given system is better than blacklists as it cannot be exhaustive and lack the ability to detect newly generated malicious URLs. The proposed system not only detects but also analyses the malicious short URLs.

S. Singhal et. al.[11] in their study propose to classify malicious and benign web- sites given their Uniform Resource Locator (URL) as input. Using the provided URL: Lexical, Host-Based, and Content-Based features are collected for the web- site. These features are fed into a supervised Machine Learning algorithm as input that classifies whether the URL is malicious or benign. The models are trained on a dataset consisting of multiple malicious and benign URLs. Further, accuracy of classification is evaluated for the following models: Random forests, Gradient Boosted Decision Trees and Deep Neural Network classifiers. Finally, a paradigm is proposed to detect and counter manually induced concept drifts. Detects dynamically modified URLs which are altered by miscreants to bypass the ML models designed to detect malicious URLs. Concept drift detection overcomes the problem of detecting and mitigating attempts by attackers for circumventing malicious website detection algorithms. Gradient Boosting model achieves a superior accuracy of 96.4%. The concept drift detection method does not detect shortened URLs. But the concept drift detection complicates the task of learning of a model.

Rishikesh Mahajan et. al.[12] in their paper implements Decision Tree, Random Forest and Support Vector Machine algorithms to detect phishing websites. Firstly, features are extracted from URL (like, presence of '@' symbol, URL redirection etc.). After feature extraction, the extracted data is fed into the above models to detect phishing websites. Feature extraction also takes factors like using of URL shortening services like TinyURL and presence of sensitive words or unicode characters, which are frequently used by malicious attackers. Random Forest model achieves a superior accuracy of 97.14% with the lowest false positive rate of 3.14%. If URLs are shortened using shortening services, then it is la- belled as malicious even though it might be benign and URLs shortened without using popular shortening services are not detected. The feature extraction does not target the network flow through the website as a parameter.

A. Singh. Et. al.[13] in their paper, implement a Multilayer Convolution Neural Network (CNN) with 2 layers. The data is first pre-processed and then tokenized. Then it is passed through an embedding layer which converts the padded input into vector form. After generation of embedded matrix for input URL, convolution operations are performed using ReLU activation function. The several generated features are then pooled to extract essential features and then it is flattened to convert entire pooled feature map matrix into a single column. Finally, it is sent to output layer which has Sigmoid and SoftMax activation functions. The model achieves an accuracy of 91% with 90% precision. The proposed model fails to detect shortened form of malicious URLs.

A. Lakshamanrao et. al.[14] in their study carry out the pre-processing using stemming and removing stop words and special symbols. Text tokenization techniques are applied on the cleaned text using Count vectorizer, Hashing vectorizer and IDF vectorizer. Finally, the tokenized text is trained using four ML classifiers Logistic Regression, KNN, Decision Tree and Random Forest. A web application is built using Flask for malicious URL detection. The study has used stemming to preprocess and clean the data. The data is also tokenized using vectorizers. Random Forest model had the best accuracy of 97.5% with Hash vectorizer. As the paper also implements Flask-based web app so it can be attached to the web browser as an extension. Special symbols are removed in the pre-processing stage as it is useful to detect whether the URL is benign or malicious. Factors like HTML tags and presence of unicode characters are not taken into consideration. Shortened URLs are not tackled and hence can bypass the model.

Bo Wei et. al.[15] in their paper, proposes a light-weight deep learning algorithm to detect malicious URLs and enable a real-time and energy-saving phishing detection sensor. This paper proposes a novel method of detecting malicious URLs using character-level multi-spatial DL model. The study also integrates the pro- posed model in an energy-saving phishing sensor. Both the URLs to be classified and the labeled URLs are fed to the model which sanitizes the input and tokenizes it before passing it to the Deep Neural Network (DNN) model. The DNN structure has six layers namely, embedding layer (Tokenizes and returns a vector), Convolutional layers, Concatenation layer which concatenates features from the previous layers, Dropout layer to prevent overfitting during training, Dense layers with ReLU activation function in each layer for extracting important features and Sigmoid layer to detect whether the URL is benign or malicious (i.e., 0 or 1). The model achieved an 86.63% true detection rate. Concatenation increases accuracy by 3% and since, the system uses character-level tokenization it detects mimic information which is commonly used by attackers. Phishing detection sensor is energy efficient it can run even run systems with low processing power and reduces the execution time by 30%. This paper discusses about building a light and efficient model which helpful for our research.

E. S. Gualberto et. al.[16] in their paper, proposes a feature engineering process- based approach on natural language processing (NLP), lemmatization, topics modeling, improved learning techniques for resampling and cross validation and hyper parameters configuration. The first proposed method uses all the features obtained from Document-Term Matrix (DTM) while the second one uses Latent Dirichlet Allocation (LDA) to deal with the problem of dimensionality. This pro- posed approach reaches a F1-measure of 99.95% using XGBoost algorithm. This model solves the three common problems faced by phishing detection models i.e., dimensionality, sparsity, and context portion. These problems are solved by statistical measures, feature selection and distributive models. After preprocessing the data, lemmatization is conducted using WordNet lexical database and then it is extracted in DTM from Bag of Words (BoW) model. The data obtained from this is used in two different fronts: directly as classification algorithms features attributes and as an input for LDA. Lastly, it is fed into the same algorithm with these two different sets of features.

Adebowale et. al.[17] in their paper, proposes a deep learning-based phishing detection solution that detects malicious websites using URL and website content like images, text, and frames. Convolutional Neural networks (CNN) and long short-term memory (LSTM) were user to build a hybrid classification model named intelligent phishing detection system (IIPS). Sensitivity of proposed model was determined from factors like, type of feature number of misclassifications and split issues. This hybrid classification solves the problem of large dataset and higher classifier prediction performance. The model achieved an accuracy of 93.28% and an average detection time of 25s. Our paper has taken references of CNN model from this paper. As this paper extracts images, it is computationally intensive and cannot be used in system with low processing power.

Routhu Srinivas Rao et. al [18] in their study proposes an application named Jail- Phish which increases the accuracy of search engine-based techniques with an ability to detect Phishing Sites Hosted on Compromised Servers (PSHCS) and also detect newly registered legitimate sites. Jail-Phish is divided into six stages they are: Extraction of domain and title, search query string preparation, search processing, initial decision making, similarity computation and final decision making. The proposed application has an accuracy of 98.61% and true positive rate of 97.77% with less than 0.64% false positive rate. This overcomes the limitation faced by search engine of detecting phishing page in compromised website and wrongly classifying of newly created or less popular domains. This paper helped us to improve our true negative rate and classify the above-mentioned web-sites correctly.

Ozgur et. al.[19] in their paper, the authors propose a real-time anti-phishing sys- tem, which uses seven different classification algorithms and natural language processing (NLP) based features. The pre-processing module firsts checks for special characters and then parses the URL for keywords. If keyword size is greater than 7, it is checked in Word Decomposer Module (WDM) to separate the sub- words they contain and are sent to the Maliciousness Analysis Module (which checks for typo squatting). The Random Word Detection Module (RWDM) detects random words using Markov Chain Model. The seven models used in this study are Naïve Bayes, KNN, Random Forest, Sequential Minimum Optimization (SMO), Adaboost, K-star and Decision Tree. Best result is obtained by Random Forest with NLP based features with an accuracy of 97.98%. This proposed architecture is language independent, real-time execution, detects new malicious web- sites based on NLP and word vector, independent from third-party services and uses feature-rich classifiers. This paper helped us choose random forest model for our research paper.

Wang et. al.[20] in their paper proposes a fast-phishing website approach called PDRCNN that only requires the URL of website. It encodes the URL into a two- dimensional tensor and feeds the tensor into a deep learning neural network to classify it. First bidirectional LSTM is used to extract global features of constructed tensor. After that, CNN is used automatically judge which character plays main role in phishing detection, capture key components of URL, and compress the extracted features into a fixed length vector space. PDRCNN achieves better performance when the above two methods are added, has a detection accuracy of 97% with average time per URL detection of 0.4ms. First length of URL string is processed and fixed to size 255 characters and then using word2vec method it is encoded into 64-bit word vector. This fed to combined model of CNN and RNN (Recurrent Neural Network) and lastly, fully connection layer and sigmoid function is used to distinguish the URL as benign or malicious. PDRCNN will not be able to classify correctly if the URL does not have relevant semantics.

2.2 Existing System

The existing system for detecting malicious websites is not comprehensive enough. Many are found to be vulnerable to zero-day exploits and do not encompass the threat of shortened URLs, and lack of user accessibility of the solution in real time. Some of the blacklisting repository methodology has the disadvantage of not being updated regularly or taking long to update this database. Many of the models rely only on the lexical features and do not encompass other features for highlighting the malicious URL. The proposed approach of this study includes extraction of lexical, web content-based and site popularity-based features.

III.PROPOSED ARCHITECTURE

3.1 Data Gathering

The dataset used in the study is a custom developed dataset made by combination of ‘ISCX-URL-2016’, ‘Malicious URL dataset’ and ‘Popular websites across the globe’ datasets. All three datasets were referred from Kaggle Dataset repository.

The Malicious URL Dataset has over 411247 unique URLs and are split into binary classes of good and bad (malicious or not). 82% of the URLs are good whilst the others are bad. The ISCX-URL-2016 dataset has over 651191 URLs in all and are split into 4 classes namely safe, 428103 in number, defacement, 96457 in number, phishing, 94111 in number, malicious, 32520 in number. The popular websites across the globe contains 9540 rows, which are top 50 websites from 191 countries. They are ranked based on trustworthiness, suggesting the user if the website which the URL is pointing to is trustworthy or not. The trustworthiness is measured by labels like Excellent, Good, Unsatisfactory, Unknown, Poor, Very Poor, etc. In total 1081195 URLs were gathered for analysis.

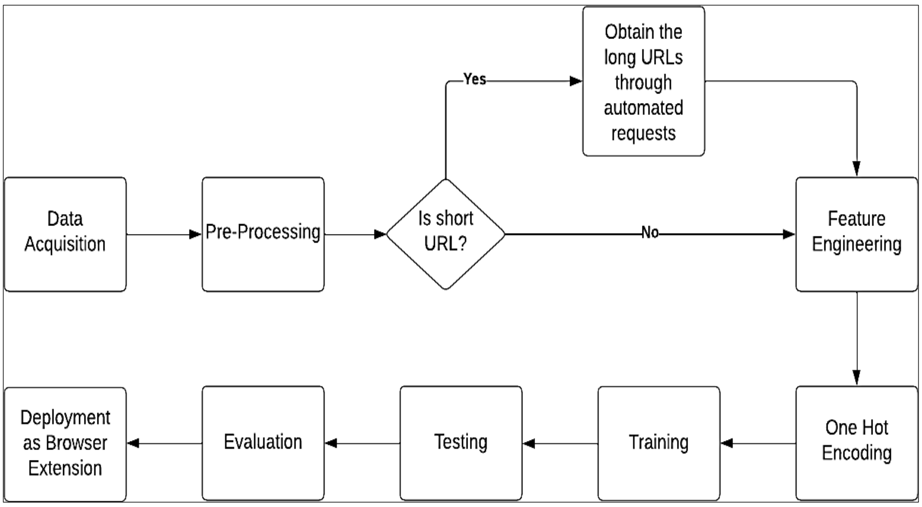


Figure. 3. Process Architecture

3.2 Data Preprocessing

Class labels have been homogenized into ‘Good’ and ‘Bad’ for this study, for ease of combination of these three datasets. Alongside this, due to combination of datasets, the possibility of redundant data values also arises. Cleaning the data and dropping redundant features was done to improve the integrity of the data gathered.

Handling short URLs is another critical step in this study. Attackers commonly using short URLs to mask the true destination of the URL, use shortening services like bitly, goo.gl, tinyURL etc. If a short URL is found, detected based on its length, the original lengthened form of the URL is obtained and then analyzed. This step in preprocessing ensures that shortened and lengthened version of the URL both are taken into consideration.

3.3 Feature Extraction

Feature Engineering and extraction is a key step for analysis and classification of malicious URLs. Lexical Features, Web Content-based features, and Site Popularity-based features are taken into consideration.

Table 1. List of Lexical Features

S.No	**Features Name**	**Description**
1	Length Of URL	Length of the URL
2	Hostname	Length of hostname
3	Suspicious Strings	Whether URL has “@”, “//‘, “?”, “=”,“-”,“_”

4	Number of subdomains	Number of dots in domain
5	Suspicious Words	Whether URL has suspicious terms or words used
6	Number of "/"	Number of backslash in URL
7	Port Number	Whether Well-known port numbers for HTTP or HTTPs
8	Path	Number of backslash in URL

***3.3.**1 Lexical Features*

Lexical features refer to the elements of the URL string or name that are defined by how they appear to users. These features are based on the visual and textual characteristics of the URL and are used to differentiate between benign and malicious links [7]. Table 1 provides a list of lexical used in this study to engineer features from the URL.

3.3.2 Web Content-Based Features

Content-based features refer to the characteristics of a website that are obtained when the site is opened or downloaded. HTML features are derived from the structural data in the HTML code as well as the content of the complete destination webpage. Based on variables including number of tags, the presence of iframes, and number of hyperlinks. We also include features of the destination webpage's Javascript, such as the number of evals, underscapes, escapes, links, execs, searches, and the presence of exception handling.

3.3.3 Site Popularity-Based Features

The core idea of site popularity features resides in the logic that the highly popular websites are non-malicious in nature. To handle this category of feature, inclusion of the dataset 'Popular Websites across the globe' was done. This step helps the analysis of websites in realtime also.

3.4 Model Development

The usage of ensemble learning was done to combine Random Forest, K Nearest Neighbors and Logistic Regression, three of the wide used binary classifiers, into a stacked model, to help classify the URLs as malicious or benign. Gradient Boosting was also applied on the model, so as to increase the accuracy and re- work on the errors of the earlier models. This stacked model was compared with its base models, alongside simple Artificial Neural Network and Decision tree for comparison. Evaluating the performance of all the models, using the performance metric of accuracy and low false positive rate, the developed stacked model was chosen, after achieving accuracy of 92%, more than the other models.

3.5 Model Deployment – Browser Plugin

A browser plugin was developed for the purpose of facilitating this solution. The architecture of the deployment can be found in Fig. 4. The browser will contain the developed plugin, using Javascript and Flask. It will read all the URLs present on the particular webpage.

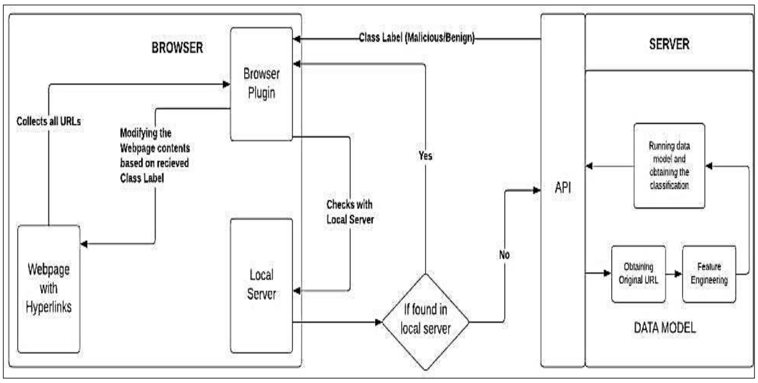


Figure. 4. Plugin Architecture

For easy accessibility and readability for the user, the plugin will automatically flag all the URLs red and green, symbolizing red meaning dangerous or malicious URL and green meaning benign URL. This flagging will take place, whence the plugin checks if the URL scanned is present in the local cache or no. If not found, the URL will be sent to the server for analysis, where the model will classify it and return the label as malicious/benign back to the plugin. Making the plugin robust and enriching the security of the process, the plugin also has a feature where it will not allow the user to click on red marked URLs directly.

IV.RESULTS AND DISCUSSION

4.1 Model Evaluation

Table 2. Comparison of all the models trained on the dataset

Model	**Accuracy**	**False Positives**	**Time to train (s)**	**Time to predict (s)**
Random Forest	90%	3e+03	1377	149
Custom Neural Network	89%	3.1e+-03	3103	256
KNN	88%	3.1e+03	304	464
Decision Tree	89%	3.1e+03	428	345
Logistic Regression	89%	2.73e+03	396	120
Stacked Model	92%	2.4e+0.3	1200	55

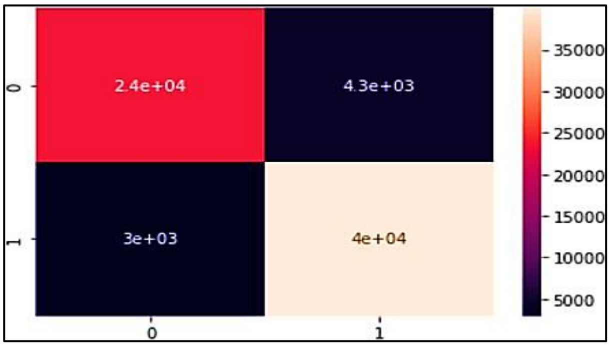


Figure 5. Confusion Matrix of Random Forest

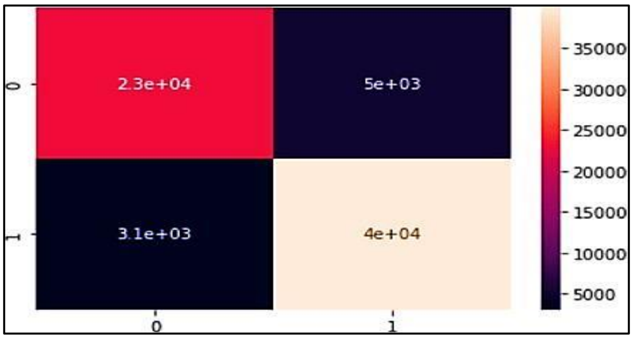


Figure. 6. Confusion Matrix of ANN

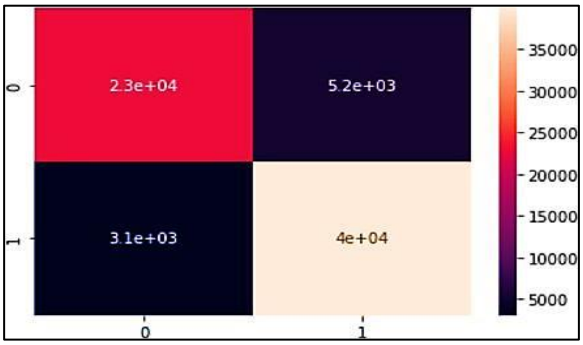


Figure. 7. Confusion Matrix of KNN

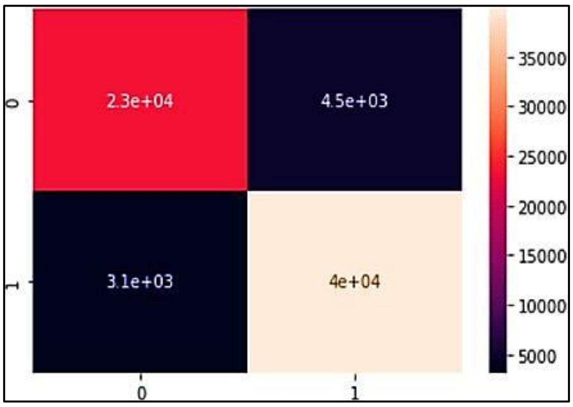


Figure. 8. Confusion Matrix of Decision Tree

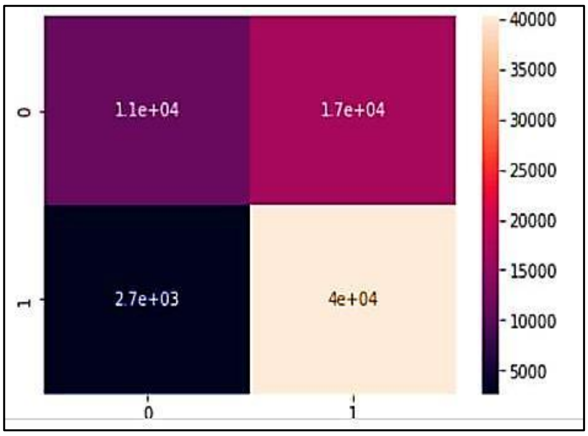


Figure. 9. Confusion Matrix of Logistic Regression

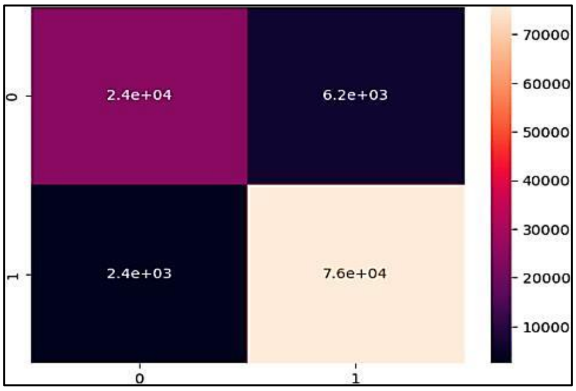
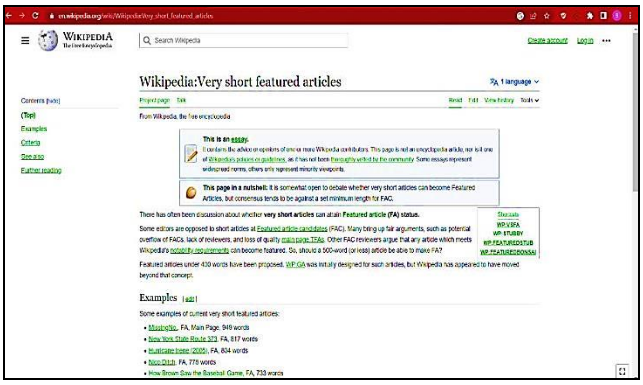


Figure. 10. Confusion Matrix of Stacked Model

4.2 Browser Plugin



en.wikipedia.org/wiki/Wikipedia:Very_short_featured_articles

Search Wikipedia

Create account Log in

Wikipedia:Very short featured articles

1 language

Project page Talk | Read Edit View history Tools

Contents [hide]

- (Top)
- Examples
- Criteria
- See also
- Further reading

From Wikipedia, the free encyclopedia

This is an essay.
It contains the advice or opinions of one or more Wikipedia contributors. This page is not an encyclopedia article, nor is it one of Wikipedia's policies or guidelines, as it has not been thoroughly vetted by the community. Some essays represent widespread norms, others only represent minority viewpoints.

This page in a nutshell: It is somewhat open to debate whether very short articles can become Featured Articles, but consensus tends to be against a set minimum length for FAC.

Shortcuts
WP:VSFA
WP:STUBBY
WP:FEATUREDSTUB
WP:FEATUREDBONSAI

There has often been discussion about whether **very short articles** can attain **Featured article (FA) status.**

Some editors are opposed to short articles at Featured article candidates (FAC). Many bring up fair arguments, such as potential overflow of FACs, lack of reviewers, and loss of quality main page TFAs. Other FAC reviewers argue that any article which meets Wikipedia's notability requirements can become featured. So, should a 500-word (or less) article be able to make FA?

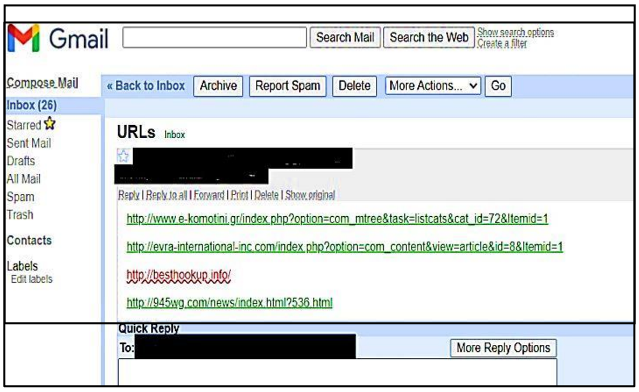
Featured articles under 400 words have been proposed. WP:GA was initially designed for such articles, but Wikipedia has appeared to have moved beyond that concept.

Examples [edit]

Some examples of current very short featured articles:

- MissingNo., FA, Main Page, 949 words
- New York State Route 373, FA, 817 words
- Hurricane Irene (2005), FA, 804 words
- Nico Ditch, FA, 778 words
- How Brown Saw the Baseball Game, FA, 733 words

Figure. 11(a). Working of Plugin in Real Time



URLs Inbox

[redacted]

Reply | Reply to all | Forward | Print | Delete | Show original

http://www.e-komotini.gr/index.php?option=com_mtree&task=listcats&cat_id=72&Itemid=1

http://evra-international-inc.com/index.php?option=com_content&view=article&id=8&Itemid=1

~~http://besthookup.info/~~

http://945wg.com/news/index.html?536.html

Quick Reply

To: [redacted]

Figure. 11(b). Working of Plugin in Real Time

V.CONCLUSIONS

Analyzing Table 2, we can conclude that the designed stacked model, developed using ensemble learning definitely outperforms its base models and other models working at an individual level. With both the aspects of performance measures, i.e, Accuracy and False Positive, the designed model performs better. With Accuracy of 92% and lowest false positive count of 2.4e+03, the model is chosen for deployment.

As seen in Section 4, regarding the results of the plugin, which ultimately leads to the facilitation of the solution for the users to protect themselves from Phishing and other attacks carried out by Malicious URLs. There is still much room for exploration and advancement in the field, as discussed in the following section, Section 6, on Future Works.

VI.FUTURE WORKS

Based on the conclusions of our study, there are several areas where future research could improve the proposed schema. In particular, we see three such areas:

1. More robust methods can be developed using secure content management for the identification of malicious URLs.
2. Performance Metrics of the model can be improved by using enhanced modalities like Ensemble Learning and Other Deep Learning Techniques.
3. Browser Extension Product can be enhanced more with action-taking features like blockage of the site if found to be malicious and reporting to authorities or addition of the site to Malicious URL Database/Repositories

REFERENCES

[1] S. R. A, M. R, R. N, S. L and A. N, "Survey on Malicious URL Detection Techniques," 2022 6th International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, 2022, pp. 778-781, doi:

10.1109/ICOEI53556.2022.9777221.

[2] Jain, A.K., Gupta, B.B. (2018). PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning. In: Bokhari, M., Agrawal, N., Saini, D. (eds) Cyber Security. Advances in Intelligent Systems and Computing, vol 729. Springer, Singapore. https://doi.org/10.1007/978-981-10-8536-9_44.

[3] Rastenis, J.; Ramanauskaitė, S.; Janulevičius, J.; Čenys, A.; Slotkienė, A.; Pakrijauskas, K. E-mail-Based Phishing Attack Taxonomy. Appl. Sci. 2020, 10, 2363. https://doi.org/10.3390/app10072363.

[4] "Phishing activity trends report 3rd quarter 2022", 2022, [online] Available: https://apwg.org/trendsreports/.

[5] Venkatesh, R., Rout, J.K., Jena, S.K. (2017). Malicious Account Detection Based on Short URLs in Twitter. In: Lobiyal, D., Mohapatra, D., Nagar, A., Sahoo, M. (eds) Proceedings of the International Conference on Signal, Networks, Computing, and Systems. Lecture Notes in Electrical Engineering, vol 395. Springer, New Delhi. https://doi.org/10.1007/978-81-322-3592-7_24.

[6] D. Antoniades, I. Polakis, G. Kontaxis, E. Athanasopoulous, S. Ioannidis, E. P.Markatos, and T. Karagiannis, “we.b: The web of short URLs,” in WWW, 2011, pp. 715–724.

[7] Manan, W.N.W., Ahmed, A.G.A., Kahar, M.N.M. (2019). Characterizing Current Features of Malicious Threats on Websites. In: Vasant, P., Zelinka, I., Weber, GW. (eds) Intelligent Computing & Optimization. ICO 2018. Advances in Intelligent Systems and Computing, vol 866. Springer, Cham. https://doi.org/10.1007/978-3-030-00979-3_21.

[8] R. Masri and M. Aldwairi, "Automated malicious advertisement detection using VirusTotal, URLVoid, and TrendMicro," 2017 8th International Conference on Information and Communication Systems (ICICS), Irbid, Jordan, 2017, pp. 336-341, doi: 10.1109/IACS.2017.7921994.

[9] D. Liu and J. -H. Lee, "CNN Based Malicious Website Detection by Invalidating Multiple Web Spams," in IEEE Access, vol. 8, pp. 97258-97266, 2020, doi: 10.1109/ACCESS.2020.2995157.

[10] Pattewar, T., Mali, C., Kshire, S., Sadarao, M., Salunkhe, J., & Shah, M. A. (2019). “Malicious Short URLs Detection: A Survey”, International Research Journal of Engineering and Technology (IRJET), 06(11), 286. https://www.irjet.net/archives/V6/i11/IRJET-V6I1162.pdf.

[11] S. Singhal, U. Chawla and R. Shorey, "Machine Learning & Concept Drift based Approach for Malicious Website Detection," 2020 International Conference on COMmunication Systems & NETworkS (COMSNETS), Bengaluru, India, 2020, pp. 582-585, doi: 10.1109/COMSNETS48256.2020.9027485.

[12] Mahajan, R., & Siddavatam, I. (2018). “Phishing Website Detection using Machine Learning Algorithms”. International Journal of Computer Applications, 181(23), 45. https://www.researchgate.net/publication/328541785.

[13] Singh and P. K. Roy, "Malicious URL Detection using Multilayer CNN," 2021 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies (3ICT), Zallaq, Bahrain, 2021, pp. 340-345, doi: 10.1109/3ICT53449.2021.9581880.

[14] Lakshmanarao, M. R. Babu and M. M. Bala Krishna, "Malicious URL Detection using NLP, Machine Learning and FLASK," 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES), Chennai, India, 2021, pp. 1-4, doi: 10.1109/ICSES52305.2021.9633889.

[15] Wei, Bo, Rebeen Ali Hamad, Longzhi Yang, Xuan He, Hao Wang, Bin Gao, and Wai Lok Woo. 2019. "A Deep-Learning-Driven Light-Weight Phishing Detection Sensor" Sensors 19, no. 19: 4258. https://doi.org/10.3390/s19194258.

[16] E. S. Gualberto, R. T. De Sousa, T. P. De B. Vieira, J. P. C. L. Da Costa and C. G. Duque, "From Feature Engineering and Topics Models to Enhanced Prediction Rates in Phishing Detection," in IEEE Access, vol. 8, pp. 76368-76385, 2020, doi: 10.1109/ACCESS.2020.2989126.

[17] Adebowale, M.A., Lwin, K.T. and Hossain, M.A. (2020), "Intelligent phishing detection scheme using deep learning algorithms", Journal of Enterprise Information Management, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/JEIM-01-2020-0036.

[18] Routhu Srinivasa Rao, Alwyn Roshan Pais, “Jail-Phish: An improved search engine

[19] SalemJeyaseelan, WR., Madhumitha, R., Yuvaraj., D.,(2019). Enhanced RSA Encrypted AODV Routing Protocol for MANET, Bioscience Biotechnology Research Communications, 12, 91-95.

[20] Bilal,Hikmat.,Sivaram,M., Yuvaraj,D.,(2019). An Improved Novel ANN Model for Detection of DDoS Attacks On

Networks, International Journal of Advanced Trends in Computer Science and Engineering, 8((1.4), 9-16.

[21] Sambandam, P., Yuvaraj, D., Padmakumari, P., & Swaminathan, S. (2023). Deep attention based optimized Bi-LSTM for improving geospatial data ontology. Data & Knowledge Engineering, 144, 102123.

[22] D. Yuvaraj, & Shuib Basri. (2022). Prevention, Reduction and Recognition of Wormhole Attack by Coordination in the Wireless Adhoc Network. IIRJET, 4(3).

[23] Manikandan.V., Yuvaraj, D.,(2019). Electrical Energy Conservation and Energy Management System Using Internet of Things, Journal of Advanced Research in Dynamical & Control Systems, VoL10., No.14, 2016-2023.

[24] B. Ahamed, R. M. S. Najimaldeen and Y. Duraisamy, "Enhancement Framework of Semantic Query Expansion Using Mapped Ontology," 2020 International Conference on Computer Science and Software Engineering (CSASE), Duhok, Iraq, 2020, pp. 56-60.

[25] Porkodi, M. V., Yuvaraj, D., Mohammed, A. S., Manikandan, V., & Sivaram, M. (2019). Prolong the Network Lifespan of Wireless Sensor Network by Using HPSM. International Journal of Mechanical Engineering and Technology, 10(01), 2039-2045.

[26] Sivaram, M., Yuvaraj, D., Porkodi, V., & Manikandan, V. (1941). Emergent news event detection from Facebook using clustering. Journal of Advanced Research in Dynamical and Control Systems, Pages, 1947, 2018.

[27] Sivaram, M., Shanmugapriya, S., Yuvaraj, D., Porkodi, V., Akbari, A., Hashim, W., & Huda, M. (2020). Decision Support System for Determining Academic Advisor Using Simple Additive Weighting. In Cognitive Informatics and Soft Computing (pp. 149-156). Springer, Singapore.

[28] Ahamed, B. B., & Yuvaraj, D. (2019). Dynamic Secure Power Management System in Mobile Wireless Sensor Network. In International Conference on Intelligent Computing & Optimization (pp. 549-558). Springer, Cham.

[29] Venkateasan, R., Yuvaraj, D.,(2018). Predicting Students' Academic Drop Out and Failures Using Data Mining Techniques, International Journal of advance Science and Technology, 28(2), 182-193.

[30] Bilal, Hikmat., Sivaram, M., Yuvaraj, D., (2019). An Improved Novel ANN Model for Detection of DDoS Attacks On Networks, International Journal of Advanced Trends in Computer Science and Engineering,8((1.4),9-16.

[31] Sivaram, M., Yuvaraj, D., Mohammed, A. S., Manikandan, V., Porkodi, V., & Yuvaraj, N. (2019). Improved Enhanced Dbtma with Contention-Aware Admission Control to Improve the Network Performance in Manets. CMC-Computers Materials & Continua, 60(2), 435-454.

[32] Priya, S. S., Yuvaraj, D., Murthy, T. S., Chooralil, V. S., Krishnan, S. N., Banumathy, P., & SundaraVadivel, P. (2022). Secure Key Management Based Mobile Authentication in Cloud. Computer Systems Science & Engineering, 43(3).

[33] Yuvaraj, D., Anitha, M., Singh, B., Karyemsetty, N., Krishnamoorthy, R., & Arun, S. (2022, October). Systematic Review of Security Authentication based on Block Chain. In 2022 3rd International Conference on Smart Electronics and Communication (ICOSEC) (pp. 768-771). IEEE.

[34] Yuvaraj, D., Kumar, V. P., Anandaram, H., Samatha, B., Krishnamoorthy, R., & Thiyagarajan, R. (2022, October). Secure De-Duplication Over Wireless Sensing Data Using Convergent Encryption. In 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT) (pp. 1-5). IEEE

[35] Venu, N., Yuvaraj, D., Barnabas Paul Glady, J., Pattnaik, O., Singh, G., Singh, M., & Adigo, A. G. (2022). Execution of Multitarget Node Selection Scheme for Target Position Alteration Monitoring in MANET. Wireless Communications and Mobile Computing, 2022.

[36] Yuvaraj, D., Priya, S. S., Braveen, M., Krishnan, S. N., Nachiyappan, S., Mehbodniya, A., ... & Sivaram, M. (2022). Novel DoS Attack Detection Based on Trust Mode Authentication for IoT. Intelligent Automation and Soft Computing, 34(3), 1505-1522.

[37] Viswanathan, M., Sivaram, M., Yuvaraj, D., & Mohammed, A. S. (2018). Security and privacy protection in cloud computing. Journal of Advanced Research in Dynamical and Control Systems, 1704-1710.