Global Scientific JOURNALS

GSJ: Volume 12, Issue 4, April 2024, Online: ISSN 2523-9376
www.globalscientificjournal.com

WEBSITE PHISHING DETECTION USING MACHINE LEARNING ALGORITHM

Ufuoma Cyril Ogude.
Department of Computer Sciences,
University of Lagos, Akoka-Yaba,
Lagos, Nigeria
uogude@unilag.edu.ng

Onwuachu Uzochukwu C.
Department of Computer Science,
Imo State University,
Owerri, Imo State, Nigeria.
onwuachu.uzochuku@imsu.edu.ng

ABSTRACT

Phishing attacks cost internet users and organization billions of dollars every year and has become a rapidly growing threat in the cyberspace. It is illegal to gather sensitive information from consumers through a number of social engineering techniques such as Email, instant messaging, pop-up messages, web pages, and other forms of communication can all be used to identify phishing tactics. This work offers a model that can determine whether a URL link is legitimate or phishing. The data set used for the classification was sourced from the University of New Brunswick dataset bank, which has a collection of benign, spam, phishing, malware, and defacement URLs, as well as from an open-source service called "Phish Tank," which contains phishing URLs in multiple formats such as CSV, JSON, etc. Phishing URLs are identified using deep neural network models. This paper create a web application software that can easily identify phishing URLs from a database of more than 10,000 URLs that have been randomly selected, divided into 50% training samples and 50% testing samples, and have up to 24,442 phishing and 5000 legitimate URLs. To distinguish between legal and phishing URLs, the URL dataset is trained and tested using feature selections like address bar-based features, domain-based features, HTTPS& JavaScript-based features. The result offered a strategy for categorizing URLs into real and phishing URLs by authenticating every link that is sent to them.

Keywords: Phishing, Deep Neural Network, Cyberspace, Features extraction and communication

1.0 INTRODUCTION

The Internet, particularly social media, has become a significant component of our lives for gathering and spreading information. Pamela (2021) claims that the Internet is a network of computers that houses important data. Security measures strive significantly more to keep users' data and devices secure when they easily give away their data or access to their computers. As a result, Imperva (2021) describes social engineering (a sort of attack designed to acquire user data, such as login passwords and credit card details) as one of the most common types of social engineering assaults. When an attacker deceives a victim into opening an email, instant message, or text message that appears to be from a trusted source, the attack occurs. When the recipient clicks the link, they wrongly believe they've gotten a present and unknowingly click a harmful link, which leads to the installation of malware, the freezing of the machine during a ransom ware assault, or the release of private data.

Due to the rapid adoption of technological advancements, there has been a significant growth in computer security threats in recent years, which has also increased the vulnerability of human exploitation. Users should be informed of the methods used by phishers as well as ways to help against falling victim to phishing. As technology develops, cybercriminals' tactics get more sophisticated. There are other ways to get consumers' personal information aside from phishing. According to KnowBe4 (2021), the following methods applies:

a) Vishing (also known as voice phishing) involves the phisher calling the victim to obtain personal information regarding the bank account. The use of a fake caller ID is the most typical method of phone phishing.
b) Smishing (SMS Phishing): Smishing is the practice of sending phony messages using the Short Message Service (SMS). By delivering a link to a phishing website, it is a technique for seducing a target using the SMS text message service.
c) Ransomware: A ransomware attack is a kind of attack that denies users access to a device or data unless they pay a ransom.
d) Malvertising: Malvertising is malicious advertising that use live scripts to push unwanted material or download malware onto your machine. Exploits for Adobe PDF and Flash are the most often utilized methods in malicious advertisements.

Consequently, this poses a growing threat to both large and small businesses as well as to people. Now that criminals have access to industrial-strength services on the dark web, there are more phishing URLs and emails being sent out and, more worrisomely, they are getting better and harder to spot.

2.0 LITERATURE REVIEW

Anjum et al, (2016) published a thorough study with the title A Literature Review on Phishing Crime, Prevention Review, and Investigation of Gaps. Various reviews of prior works of literature are offered. In order to combat phishing, the report suggests using CRI, which stands for Crime, Prevention Review and Investigation of Research Gap.

Ashritha et al, (2019) reviewed Detection of Phishing Websites Using Machine Learning suggested many algorithms (models), as well as various elements of phishing assaults and strategies to detect phishing websites. The paper's discussion of works and various phishing detection techniques is one of its strong points. Additionally, it presents a suggested mechanism for precisely predicting phishing websites. The inquiry hole gives researchers additional room to explore phishing detection.
Kiruthiga. & Akila, (2019) outlined an innovative technique for employing machine learning algorithms to identify phishing websites. Additionally, they evaluated the performance of five machine learning algorithms: Generalized Linear Model (GLM), Generalized Additive Model (GAM), Gradient Boosting (GBM), Random Forest (RF), and Decision Tree (DT) (Shad & Sharma, 2018). Each algorithm's accuracy, precision, and recall assessment metrics were computed and compared. The performance of the top three algorithms, Decision Tree, Random Forest, and GBM, was compared in the table. The Random Forest algorithm yielded the maximum 98.4% accuracy, 98.59% recall, and 97.70% precision, according to the tables of accuracy, recall, and performance.

Sönmez et al. (2018), propose a categorization approach to classify phishing attacks. Website classification and feature extraction from web pages are included in this approach. The ideas for phishing feature extraction have been explained, and thirty features have been extracted from the UCI Irvine machine learning repository data set. The data was classified using these features using Support Vector Machine (SVM), Naive Bayes (NB), and Extreme Learning Machine (ELM) (Sönmez et al., 2018). The Extreme Learning Machine (ELM), which exceeded SVM and NB in accuracy with 95.34%, used six activation functions. The results were assisted by the usage of MATLAB.

Peng et al.(2018) offer a method for identifying phishing email attacks using machine learning and natural language processing. In order to find malicious intent, the text is subjected to a semantic analysis. Each sentence is parsed using a natural language processing (NLP) technique to determine the semantic roles of the words in relation to the predicate. The Nazario phishing email set dataset is utilized in conjunction with Python programs to create this technique. Comparison of Net-craft with SEAHound results (Peng et al., 2018) reveals precision of 98% and 95%, respectively.
The Table 2.1 shows related algorithms proposed by several researchers in Machine Learning to detect phishing websites. On reviewing their papers, they concluded that most of the work done is by using familiar machine learning algorithms like Naïve Bayesian, SVM, Decision Tree, and Random Forest. Some authors proposed a new system like Phish Score and Phish Checker for detection. The combinations of features with regards to accuracy, precision, recall, etc. were used. Experimentally successful techniques in detecting phishing website URLs were summarized in Table 2.1.

Table2.1 :Outline of related algorithms used to detect phishing website

Algorithm used	**Reference paper**	**No. of Features**	**Dataset**	**Language /Tools**	**Conclusion**
Decision Tree (DT), Random Forest(RF), Gradient Boosting (GBM), Generalized Linear Model (GLM), Generalized Additive Model(GAM)	Shad, and Sharma, (2018)	30	Not Mentioned	Python,R Language	Random Forest highest accuracy 98.4%
Support vector Machine (SVM),Naïve Bayes (NB) and Extreme Learning Machine(ELM)	Sönmez, et al (2018)	30	UCI-Machine Learning Repository	MATLAB	ELM achieved 95.34% accuracy.
Natural Language Processing	Peng, et al. (2018)	-	Nazario phishing Emailset	Python	Proposed SEA Hound provides 95% accuracy
Random Forest	Saimadhu. (2017)	8	Phish tank,	R Studio	95% accuracy
Neural network model Adam AdaDelta and SGD	Shreya, (2020)	URL length	Phishtank	Chainer	Accuracy of Adam 94.18%
Convolution neural network(CNN) and SNN long short-term memory(CNN-LSTM)	Kondeti et al. (2021)	-	Phishtank, Open Phish, Malware Domain list, Malware Domain	Tensor Flow in conjunction with Keras	CNN-LSTM obtained 98% accuracy
Logistic Regression and Support Vector Machine(SVM)	Noel, (2016).	19	UCI machine learning repository	BigData	SVM accuracy 95.62%

daBoost,Bagging,RandomForest, andSMO	Kartik, (2021)	11	Direct Industry Anti-Phishing Alliance of China	BigData	Only Semantic Features of word embedding obtained high accuracy.
C4.5 decisiontree	Almomani et al. (2015)	9 Features and heuristic values	Phish tank Google	-	89.40%
KNN,SVM and RandomForest	Gupta et al (2016)	22	UCI-Machine Learning Repository	HTML, JavaScript, CSS, Python	Random Forest high accuracy
Naïve BayesandSequentialMinimalOptimization(SMO)	Rishikesh & Irfan, (2018).	133	Phish tank Google	C# programming and R programming WEKA	SMOBeataccuracythanNB
Heuristic feature root mean square Error(RMSE)	Rami et al. (2015),	6	PhishTank	MYSQL.PHP	97%
Phish Score	Shaikh et al. (2016)	12	PhishTank	-	94.91%
Phish Checker	Abdelhamid et al. (2017)	5	PhishTank andYahoodirectoryset	Microsoft Visual Studio Express2013 and C# language	96%

3.0 MATERIALS AND METHODS

The new phishing detection system makes use of Random Forest, Multilayer Perceptions, Auto Encoder Neural Network, Support Vector Machine, Decision Tree, and XGBooster. These models were chosen based on several comparisons between the results of various machine learning methods. These models are each tested and trained using a website content-based feature that is taken from phishing and authentic datasets. Therefore, the most accurate model is chosen and implemented into a web application that will allow a user to determine whether a URL link is authentic or phishing

Data Collection

Different open-source platforms provide the data that is utilized to create the datasets used to train the models. The dataset collection includes both legal and phishing URL datasets. The collection of phishing URLs comes from Cisco Talos Intelligence Group's open-source Phish Tank service. This site offers a collection of phishing URLs that are updated every hour in a variety of forms, including CSV, JSON, and others. The dataset was collected from the phishtank.com website. Over 24,442 random phishing URLs are gathered from this dataset to train the ML models.

The University of New Brunswick's open datasets provide the set of legitimate URLs accessible on the university website. This dataset contains a collection of URLs that aren't malicious, spamming, phishing, or defacement. The legitimate URL dataset is taken into consideration for this study out of all of these types. Over 5000 randomly selected valid URLs from this dataset are gathered to train the ML models.

A. Preprocessing

The first and most important step after data collection is data preprocessing. By eliminating redundant and erroneous data and encoding the raw dataset for phishing detection using the One-Hot Encoding approach, the raw dataset was made ready for the machine learning model.

B. Exploratory data analysis

Following a number of data cleaning steps, the dataset was subjected to exploratory data analysis (EDA). The dataset was examined, explored, and summarized using the data visualization technique. To find patterns and insights in data, these visualizations use heat maps, histograms, box plots, scatter plots, and pair plots.

C. Feature Extraction

By extracting new features from the current ones in a dataset, feature extraction seeks to lower the overall number of features in the dataset. As a result, phishing and legitimate datasets were used to extract website content-based features, such as the address bar-based feature, which has 8 features, the domain-based feature, which has 3 features, and the HTML & JavaScript-based feature, which has 4 features. 15 features were thus extracted in total for phishing detection.

Architectural design focuses on understanding how a system should be set up and developing the overall structure of that system. It demonstrates how the system's various parts interact to accomplish its primary goals. It is the procedure for determining the various components that make up a system as well as the framework for sub-system coordination and communication. The architectural design of the suggested system is shown graphically in the diagram below. When a user enters a URL link, the link passes through several trained machine learning and deep neural network models before the best model with the highest accuracy is chosen. The chosen model is implemented as an API (Application Programming Interface) and then incorporated into a web application. As a result, a user engages with the online application, which is available on many display devices including PCs, tablets, and mobile devices. The use case scenarios for the phishing detection system are shown in Figure 1

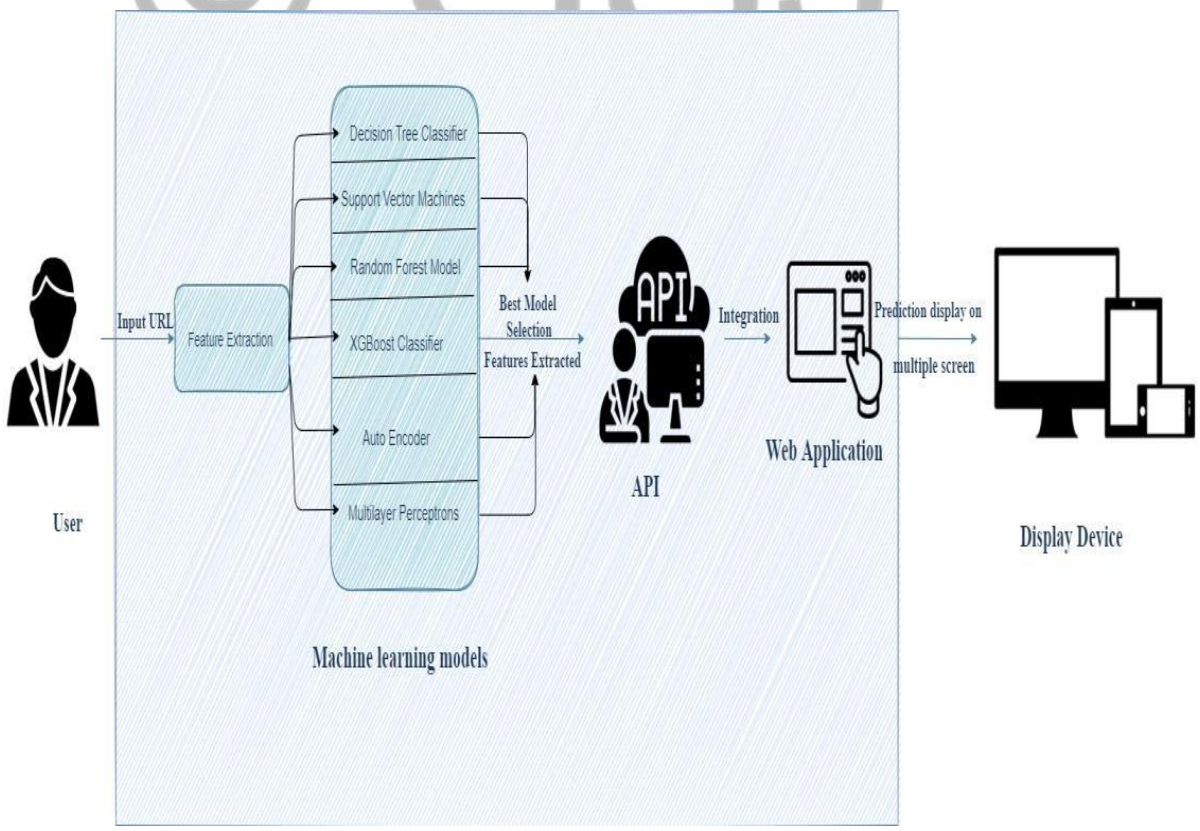


Figure 1: Architectural Design of the Proposed System

Figure 2 shows the functionality of the system as designed from the requirements is described in the use case diagram, which also provides an overview of the system's users. It represents the observable interactions between actors and the developing system as a behavior diagram. Actors, the system, associated use cases, and relationships between them are all included in the use case diagram.

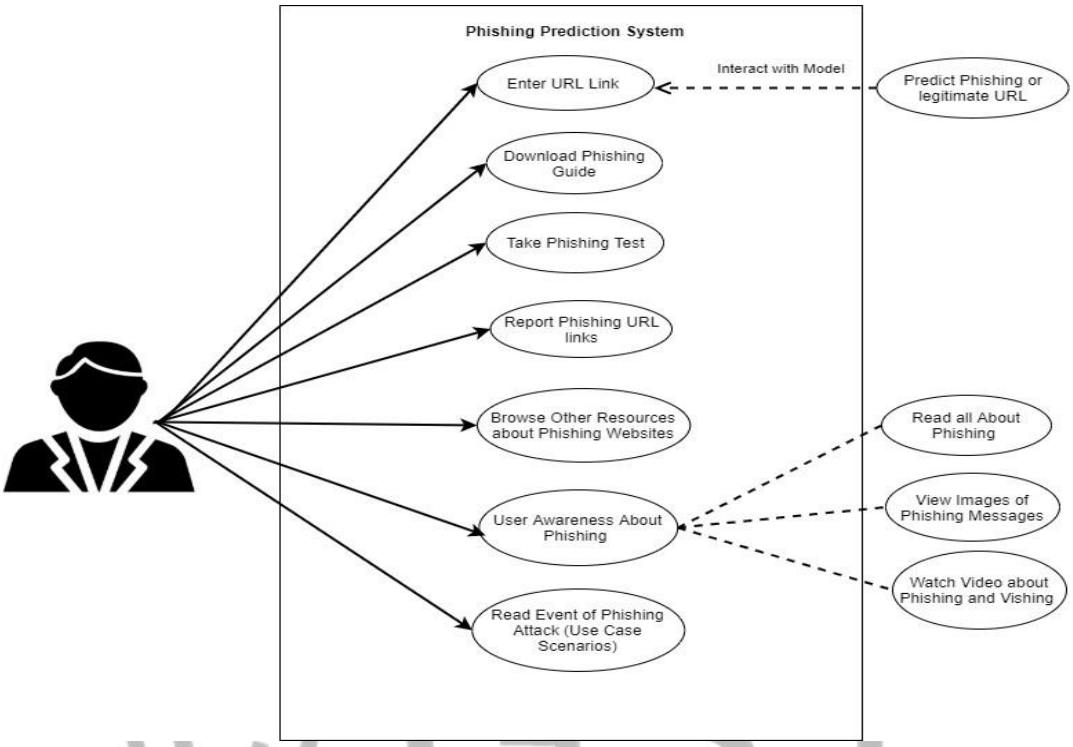


Figure 2: Use Case diagram for Proposed System

A flowchart is a diagram that shows how a system, computer algorithm, or process works. It is a graphical depiction of the system's stages to be carried out, listing them in chronological sequence. It is intended to convey complex processes in simple, understandable representations and to show how algorithms run. The machine learning technique used by phishing detection systems is depicted in Figure 3..

The phishing detection web interface system is displayed in Figure 4. When a user enters a URL link, the website analyzes the URL's format and then determines whether the link is legitimate or phishing.

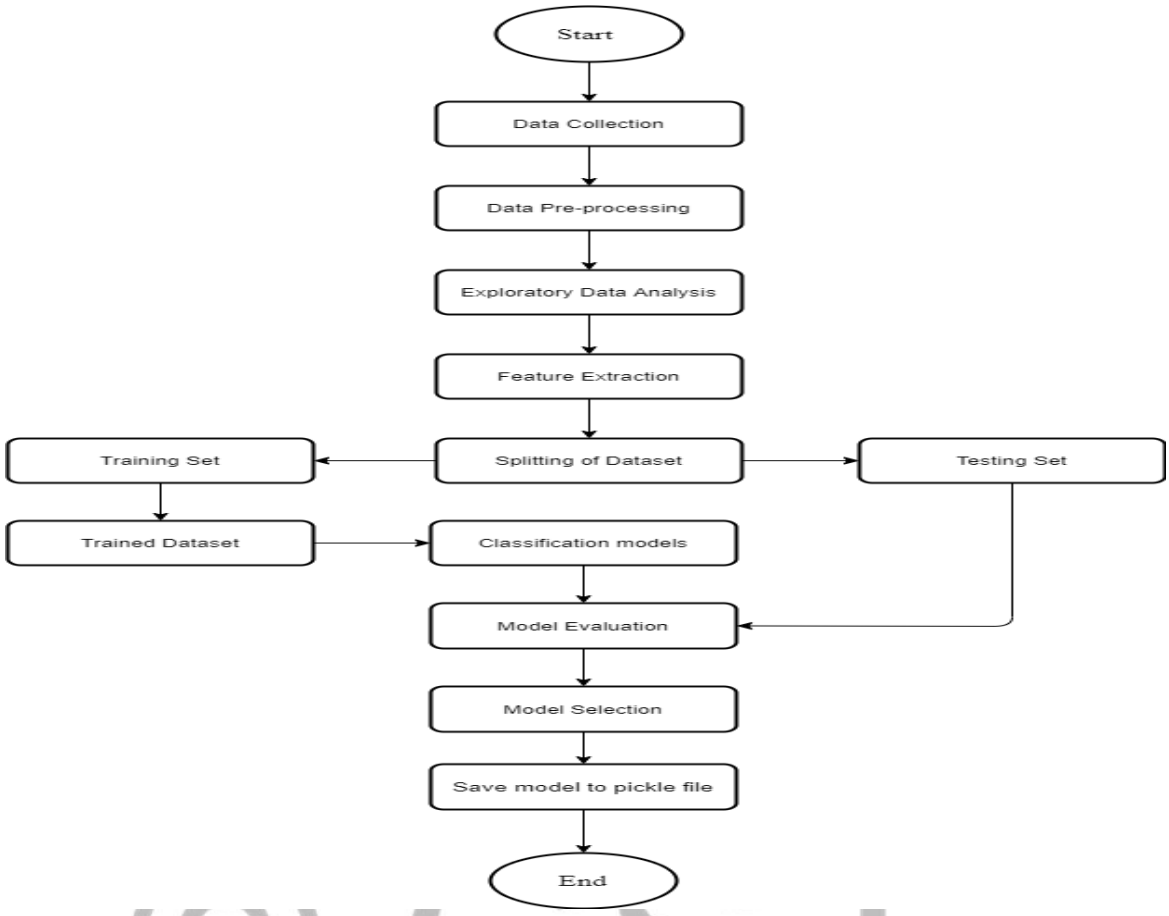


Figure 3: Flowchart of the proposed System

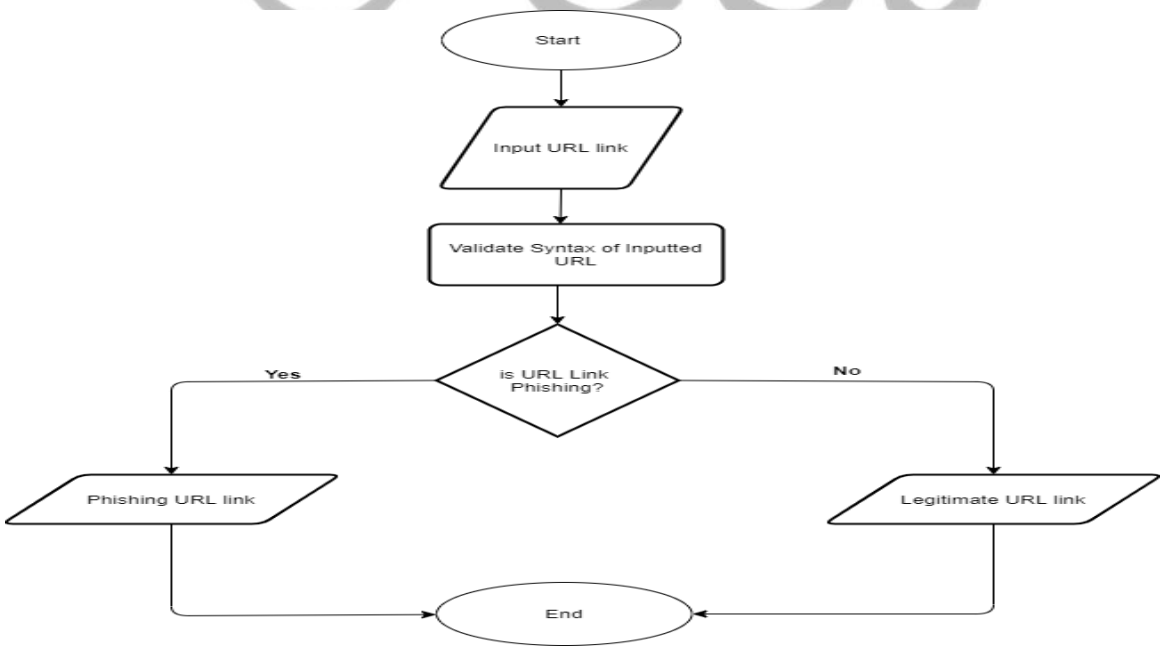


Figure 4: Flowchart of the web interface

4.0 Experiment and Results

"PHISH-BOT" is a one-page phishing detection web application can be used with any browser. Python was the only programming language used to create the application. The following pages in figure 5 are part of the phishing detecting website application:

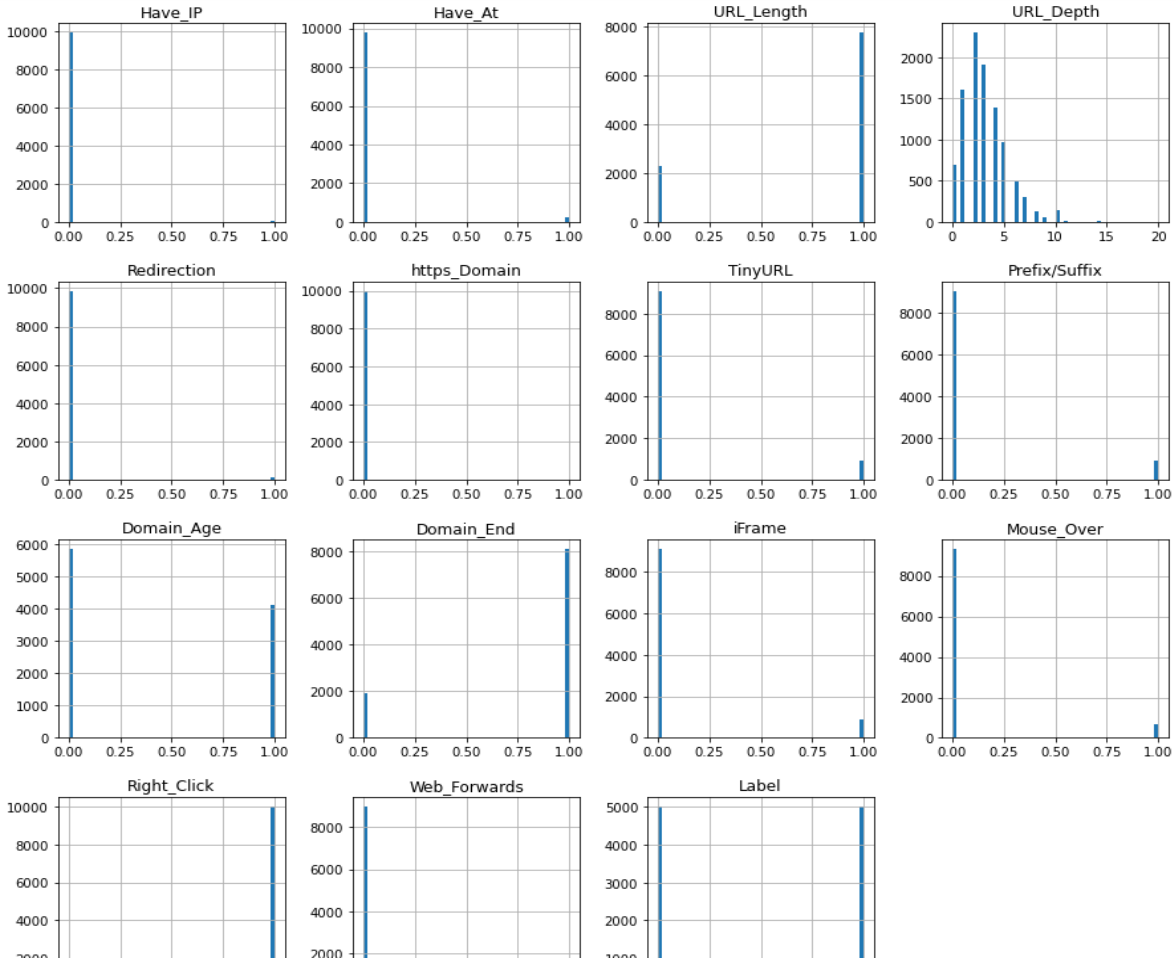


Figure 5: Dataset distribution plot based on the chosen features

On the home page, there is a session where a user can enter a URL and determine whether it is phishing or not. It forecasts the URL's current state. This page's users can use it to verify a URL link and to access a variety of phishing attack materials. To learn how to recognize phishing messages and URLs, the User can explore the resource tab.

The Predict URL page

The predict URL page as shown in figure 6(a) and figure 6(b). This is the page users will input the suspicious URL to get the prediction. The output will determine if the URL is legitimate or phishing..

Resource page

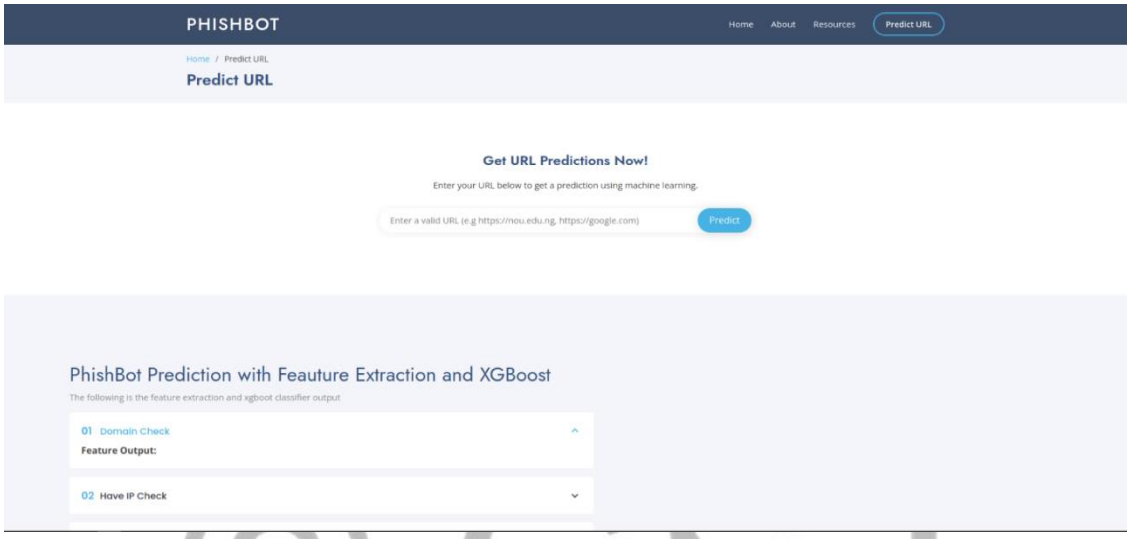
It includes many materials on phishing, including explanations of the term, types of phishing attacks, and strategies, as well as references to the sources

from which the content was gathered. Additionally, it has two (4) sub-session links: the google safe browsing, google search help, intradyn and jigsaw phish quiz as seen in Figure 7.

Web application Source Code

As seen in figure 8, the web application's source code is divided into pages and is written in python.

Figure 6. (a):The Predict URLpage



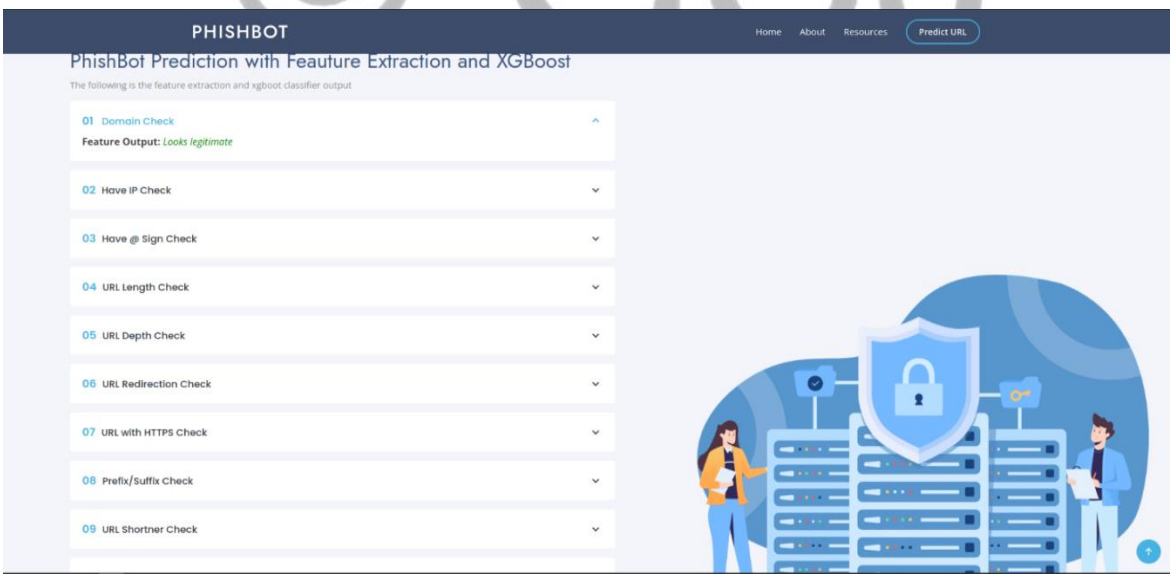


Figure 6 (b): ThePredict URLpage

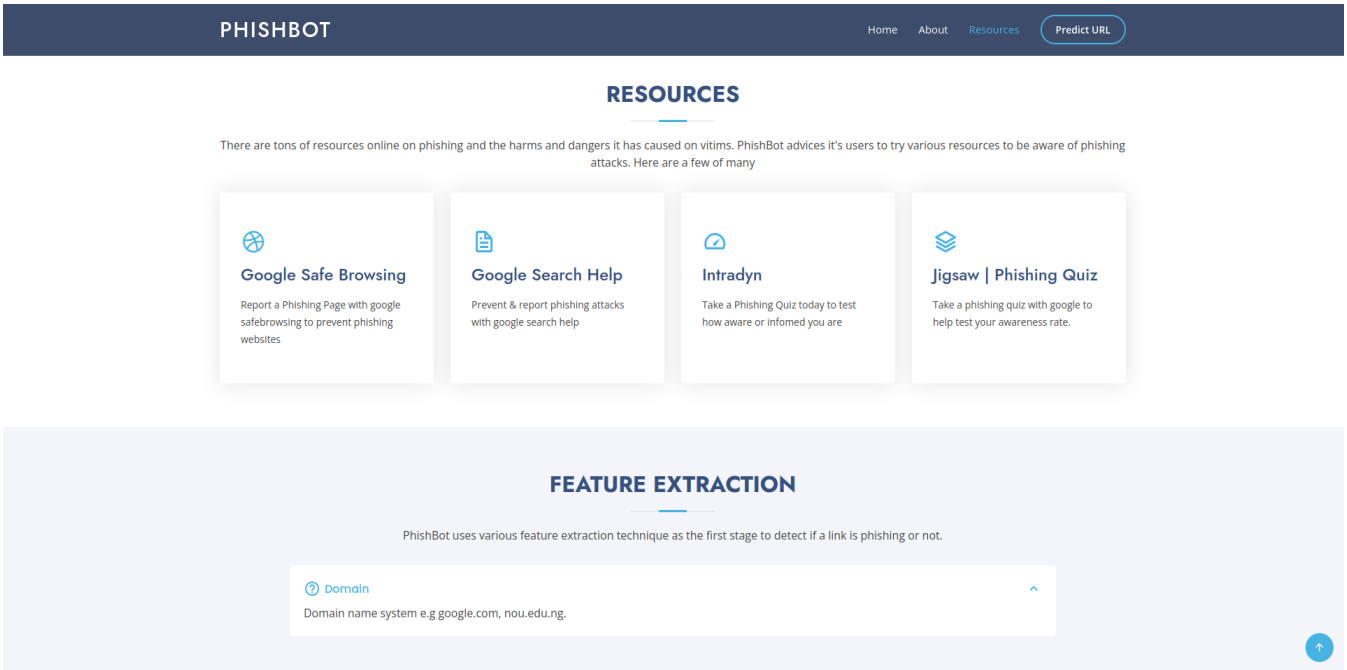


Figure 7: The Resource page

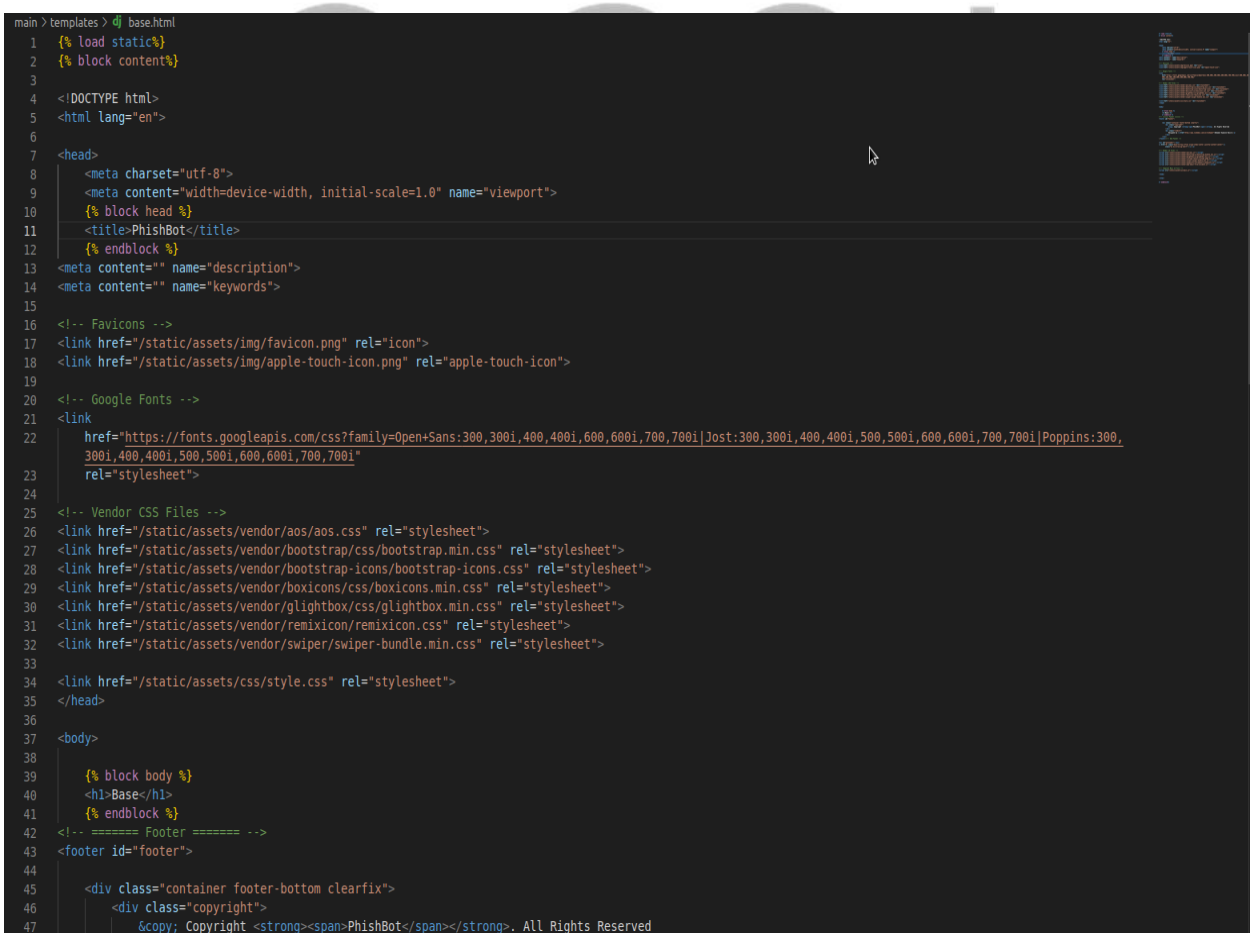


Figure8 :Code for the web application

4.0 Result Discussion

PHISH-BOT" is a one-page phishing detection web application can be used with any browser. Python was the only programming language used to create the application. The pages displayed in figure 5 are part of the phishing detecting website application: On the home page, there is a session where a user can enter a URL and determine whether it is phishing or not. The developed system can forecast the URL's current state. The page's users can use it to verify a URL link and to access a variety of phishing attack materials. To learn how to recognize phishing messages and URLs, the User can explore the resource tab. The predict URL page as shown in figure 6(a) and figure 6(b). This is the page users will input the suspicious URL to get the prediction. The output will determine if the URL is legitimate or phishing.

The new system includes many materials on phishing with the explanations of the term, types of phishing attacks, and strategies, as well as references to the sources from which the content was gathered. Additionally, it has two (4) sub-session links: the google safe browsing, google search help, intradyn and jigsaw phish quiz as seen in Figure 7. Web application Source Code as seen in figure8, the web application source code is divided into pages.

Conclusion

The developed system provides users with access to new and quicker technique to determine if a URL link is real or phishing as well as an instructional material regarding phishing attacks. It uses deep neural network methods and machine learning models to determine whether a URL link is real or phishing. Phishing URLs were specifically identified using feature extraction and models applied to the dataset, which also improved the performance accuracy of the models. It is also remarkably effective at determining whether a URL link is legitimate.

REFERENCES

Abdelhamid, N., Thabtah F., & Abdel-Jaber, H. Phishing detection: A recent intelligent machine learning comparison based on models' content and features," 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, 2017, pp.

Almomani, A., Gupta, B. B., Atawneh, S., Meulenberg, A., & Almomani, E. (2015). A survey of phishing email filtering techniques, Proceedings of IEEE Communications Surveys and Tutorials, vol. 15, no. 4, pp. 2070–2090.

Anjum N. S., Antesar M. S., & Hossain M.A. (2016). A Literature Review on Phishing Crime, Prevention Review and Investigation of Gaps. Proceedings of the 10th International Conference on Software, Knowledge, Information Management & Applications (SKIMA).

Ashritha, J. R., Chaithra, K., Mangala, K., & Deekshitha, S. (2019). A Review Paper on Detection of Phishing Websites using Machine Learning.Proceedings of International Journal of Engineering Research & Technology (IJERT), 7, 2. Retrieved from www.ijert.

Gupta, B. B., Tewari, A., Jain, A. K., & Agrawal, D. P. (2016). Fighting against phishing attacks: state of the art and future challenges, Neural Computing and Applications.

Imperva. (2021). Phishing attacks. Retrieved from https://www.imperva.com/learn/application-security/phishing-attack-scam/.

Kartik, M. (2021). Everything You Need to Know About Feature Selection in Machine Learning. Retrieved from https://www.simplilearn.com/tutorials/machine-.

Kiruthiga, R., Akila, D. (2019). Phishing Websites Detection Using Machine Learning. Retrieved from https://www.researchgate.net/publication/337049054 Phishing Websites Detection.

KnowBe4 (2021). Phishing Techniques. Retrieved from https://www.phishing.org/phishing-techniques.

Kondeti, P. S., Konka, R. C., & Kavishree, S. (2021). Phishing Websites Detection using Machine Learning Techniques. International Research Journal of Engineering and Technology, 08(4), Page 1471-1473. Retrieved from https://www.irjet.net/archives/V8/i4/I.

Noel, B. (2016). Support Vector Machines: A Simple Explanation. Retrieved from https://www.kdnuggets.com/2016/07/support-vector-machines-simple-.

Pamela (2021). Phishing attacks. Retrieved fromhttps://www.khanacademy.org/computing/computersandinternet/xcae6f4a7ff01e7d:online-data-security/xcae6f4a7ff015e7d:cyber-attacks/a/phishing-attacks

Peng, T., Harris, I., & Sawa, I. (2018). Detecting Phishing Attacks Using Natural Language Processing and Machine Learning. Proc. - 12th IEEE Int. Conf. Semant. Comput. ICSC 2018, vol. 2018–Janua, pp. 300–301.

Rami, M. M., Fadi, T., & Lee, M. (2015). Phishing Websites Features. Retrieved from https://eprints.hud.ac.uk/id/eprint/24330/6/MohammadPhishing14July2015.pdf.

Rishikesh, M., & Irfan, S. (2018). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications, 23, 45. doi:10.5120/ijca2018918026.

Saimadhu, P. (2017). How the random forest algorithm works in machine learning. *Retrieved from https://dataaspirant.com/random-forest-algorithm-machine-*.

Shad, J., & Sharma, S. (2018). A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology, pp. 425–430.

Shaikh, A.N., Shabut, A.M., Hossain, M.A. (2016, December 15-17). A literature review on phishing crime, prevention review, and investigation of gaps. Paper presented at the Tenth International Conference on Software, Knowledge.

Shreya, G. (2020). Phishing website detection by machine learning techniques. Retrieved from https://github.com/shreyagopal/Phishing-Website-Detection-by-.

Sönmez, Y., Tuncer, T., Gökal, H., & Avci, E. (2018). Phishing web sites features classification based on extreme learning machine. 6th Int. Symp. Digit. Forensics Secure. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5.