International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 11 Issue: 09 | Sep 2024 www.irjet.net p-ISSN: 2395-0072

PHISHING WEBSITE DETECTION USING MACHINE LEARNING AND DEEP NEURAL NETWORKS

Deepak Kumar Jha[1], Pallavi Mishra[2], Aradhya A Rathore[3], Sankalp Verma[4]

[1-4]*Vellore Institute of Technology, Vellore, Tamil Nadu, India*

Abstract -

In the digital world, phishing is still a prevalent and developing problem that can be extremely costly for both individuals and businesses. The ever-evolving strategies used by fraudsters make it more difficult to identify phishing assaults with high precision. As a reaction, we suggest creating an intricate phishing detection system that makes use of cutting-edge machine learning techniques. Our method looks for trends and abnormalities in large datasets to increase the efficacy and accuracy of phishing attempt detection. Our solution aims to offer a comprehensive defence mechanism that can adapt to the dynamic nature of phishing attacks, protecting users and organisations from sophisticated and newly emerging phishing schemes as digital environments get more complicated.

Key Words: Phishing Detection, Machine Learning, Cybersecurity, Fraud Prevention, Digital Threats, Attack Identification, Data Analysis, Threat Mitigation

1. INTRODUCTION

1.1 Brief History

Early in the 1990s, America Online (AOL) computers were the target of the first known phishing assault [37]. By establishing fictitious accounts with erroneous credit card information, attackers took advantage of AOL's initial validation procedure. These accounts were used to access AOL resources after they were activated. In response, AOL strengthened its verification processes; however, hackers adjusted by obtaining personal data from actual users. They obtained user credentials byimpersonating AOL staff in phishing emails and instant chats. This strategy spread to encompass a number of e-commerce and banking websites.

1.2 Statistics

2.97 billion people, or more than 38% of the world's population, were online as of 2014. These users have been the subject of phishing schemes more often, which have resulted in large financial losses. Phishing attacks increased 160% in 2012 over 2011. Approximately 450,000 phishing assaults resulted in losses over $5.9 billion in 2013. There were 125,215 attacks in the first quarter of 2014, up 10.7% from the previous quarter. 99.4% of phishing sites used port 80, and more than 55% used the name of the target. 123,972 phishing assaults were reported in the second half of 2014; as a result, 4.5 billion dollars in 2014 and 4.6 billion dollars in 2015 were lost financially.

The following image shows a simplified version of how phishing websites operate.

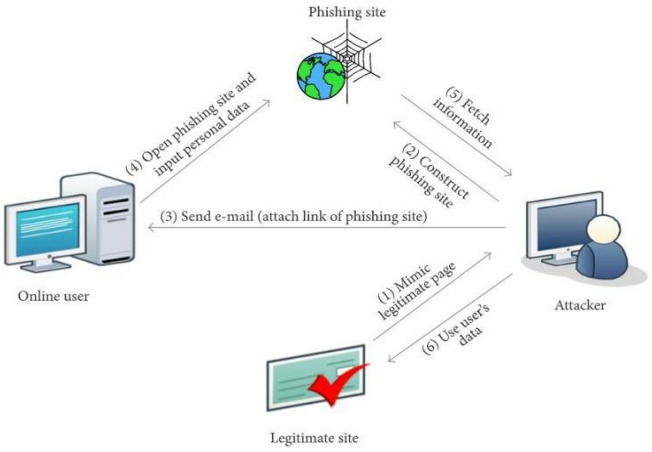

Fig-1: Illustration of a Phishing Attack Workflow

2. METHODOLOGY

2.1 Workflow

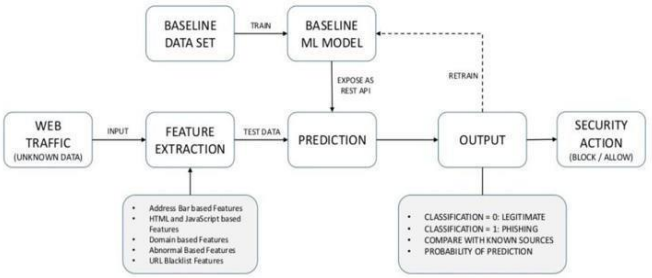

Fig-2: Flowchart of the Phishing Detection System

2.2 Module Description

2.2.1 Exploratory Data Analysis (EDA)

In this step we study the trends in data and count thenumber of observations for each class of data (i.e. whether it is a phishing website or a legitimate website).

2.2.2 Data Visualization

In this module we create simple graphs to visualize the trends in data, namely we make a heatmap which visualizes the correlation of the data features and the target class and how heavy these correlations are, further to simplify this we create a simple bar graph again showing the correlation of our data features with the target class this helps us in visually understanding which features affect the target class the most and which features are rather insignificant in the outcome of our classification.

2.2.3 Pre - Processing the data

In the dataset we are using the result class has two distinct values (-1 and 1), a machine learning classifier generally accepts the 2 values of 0 and 1 so we remap the -1 class results to 0, we also check if the datasethas any NaN values and try to fill them outusing proper techniques.

2.2.4 Train Test Validation split

In this step we split the data into two data sets whichwill be out respective training and testing data sets which will be in the ratio of 80:20 (training data : testdata)

2.2.5 Implementation Using Streamlit:

Streamlit is an open-source python library that is useful to create and share data web apps. It is slowly gaining a lot of momentum in the data science community. Because of the ease with which one can develop a data science web app, many developers use it in their daily workflow. In this final module weaim to take the best fitting module and implement ina user-friendly Web Application that any common person with minimal knowledge of machine learning or internet security can use to easily get a highly accurate prediction of whether a website is legitimate or a phishing website.

2.3 Procedure Description

In this study, we experimented with various machinelearning models, including Decision Trees and Random Forests, to identify the most suitable model for our dataset. After extensive evaluation, we selected the model that demonstrated the highest accuracy among all the algorithms tested.

Technique	Description
1.Logistic Regression	Logistic Regression uses the equation - $\ell = \beta 0 + (\beta 1)(x1) + (\beta 2)(x2)$ Where: • ℓ is the linear combinationof the input features. • β0 is the intercept (bias term). • β1,β2,…,βn are the coefficients (weights) for each feature x1,x2,…,xn. The output is passed through a sigmoid function, which returns a value between 0 and 1. A cut-off point (e.g., 0.5) classifies the resultinto binary classes. $P(y = 1) = \sigma(\ell) = 1/(1 + e^{-\ell})$ Where: • P(y=1) represents the predicted probability thatthe output class is 1. • $\sigma(\ell)$ is the sigmoid function, which squashesthe linear output into therange [0, 1].
2. K- Nearest Neighbours	A new data point is classified using the k-Nearest Neighbours (k-NN) technique by utilizing the classes ofthe k closest data points from the training set. Metrics like the Minkowski, Manhattan, and Euclidean distances are frequently used to calculate the distance between the points. The judgment is extremely susceptible to noise or outliers when k=1, as the procedure allocates the class of the closest single data point to the new instance. When k is bigger than zero, the algorithm selects the class that is most common among the k nearestpoints (majority voting). This method lessens the effect of noise

	since the algorithm considers more neighbours before reaching a conclusion. The selection of k is essential: -Smaller k values: The model will be more adaptable, but it may be unduly impacted by spurious or unimportant data items. -Large k values: The model gets more stable and broadly applicable, but it could also over smooth the borders between classes, which could lead to points close to the decision boundaries being incorrectly classified. Another option is weighted voting, in which neighbours who live closer to the new point have a greater say in how the area is classified.
3. Decision Trees	For both classification and regression tasks, a supervised learning approach called a decision tree is employed. The arrangement resembles a flowchart, wherein: Every internal node depicts a test on a feature (attribute), every branch shows the test's results, and every leaf node shows a class labelor other result. To reduce uncertainty (as assessed by metrics like Gini impurity or information gain), the tree divides the dataset into subgroups depending on the most important characteristic at each stage. Decision trees are quite versatile because they may capture non-linear correlations between characteristics and the target variable. A Random Forest is a powerful ensemble learning technique that constructs multiple decision trees during training and aggregates their predictions to make a final decision. Each tree in the forest is built using a different random subset of the training data, a process known as bagging (bootstrap aggregating). Additionally, at each node split, only a random subset of features is considered, which introduces further diversity among the trees.

4. Random Forest Classifier	The final output is determined by: • **Majority voting** for classification tasks, where the class most predicted by the individual trees is selected. • **Averaging** for regression tasks, where the mean of all tree outputs is taken as the final prediction. This combination of multiple trees reduces variance and overfitting, improving generalization to unseen data. Random Forests are also known for their ability to handle large datasets and high- dimensional spaces, offering better performance on complex problems while retaining interpretability through feature importance analysis.
5. Support Vector Classifier	In this study, Support Vector Classifiers (SVC) are employed to determine the optimal decision boundary for classifying data points. The primary objective of SVC is to maximize the margin, which is the perpendicular distance between the decision boundary (hyperplane) and the nearest data points from each class, referred to as **support vectors**. By focusing on maximizing this margin, SVC aims to enhance the model's generalization capability tonew, unseen data. The **support vectors** play a crucial role in defining the position and orientation of the hyperplane, as they are the closest data points to the decision boundary. In scenarios where the data is not linearly separable, SVC utilizes kernel functions—such as the radial basis function (RBF) or polynomial kernels—to map the data into a higher-dimensional space where a linear separation is feasible. This **kernel trick** allows SVC to effectively manage complex decision boundaries thatcannot be captured in the original feature space.

6. Deep Learning and Artificial Neural Networks	The SVC algorithm involves solving a quadratic optimization problem to find the hyperplane that maximizes the margin while ensuring accurate classification of training instances. This approach contributes to SVC's robustness in high-dimensional spaces and its ability to create effective decision boundaries, even for intricate datasets. Deep Learning is an advanced machine learning technique that utilizes neural networks with multiple layers to model complex relationships between inputs and outputs. These neural networks consist of an input layer, hidden layers, and output layer, where each layer learns to transform the data through learned weights and activation functions. • **Neural Network Structure**: Deep learning models are composed of multiple interconnected layers, which enable the network to learn hierarchical features and capture intricate patterns in data. • **Training Mechanism**: The network is trained using backpropagation and optimization algorithms, such as stochastic gradient descent (SGD), to minimize prediction errors and improve accuracy. • **Applications**: This approach is widely used in various fields, including computer vision, language translation, and speech recognition, due to its ability to effectively handle high-dimensional and unstructured data. In essence, deep learning provides powerful tools for modeling and interpreting complex data, making it a pivotal technology in modern artificial intelligence.

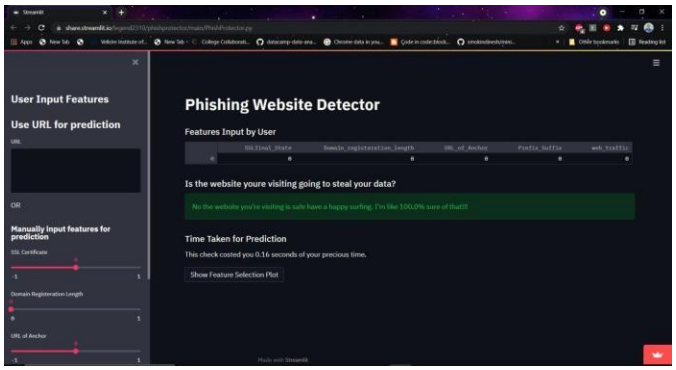

Fig-3: Website Interface

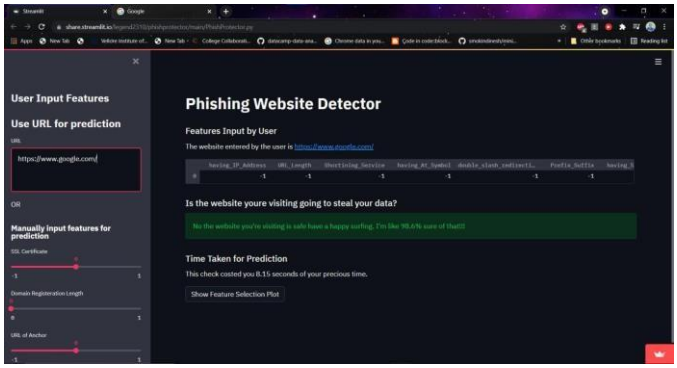

Fig-4: Sample prediction on Google.com

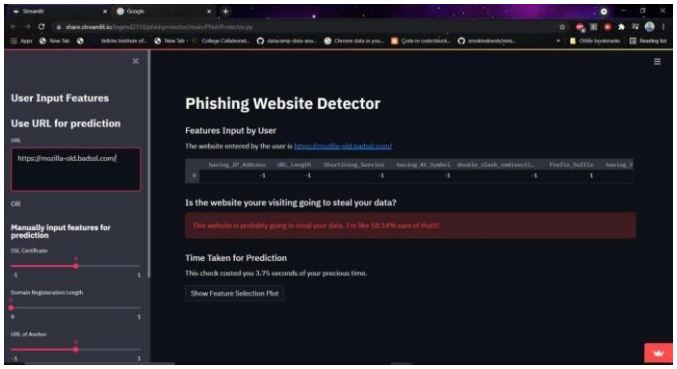

Fig-5: Sample prediction on a Phishing Website

3. CONCLUSIONS

In conclusion, our project signifies a critical step forward in combating phishing and other cyber threats. By utilizing a range of models, including Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Logistic Regression, Decision Trees, Random Forest, and Deep Neural Networks (DNN), we achieved diverse accuracy scores: SVM and Decision Trees at 94%, KNN at 91%, Logistic Regression at 92%, DNN at 95%, and Random Forest achieving the highest at 97%. This peak accuracy of 97% illustrates the efficacy of our approach in accurately identifying phishing attacks, which can play a pivotal role

in mitigating the substantial financial losses associated with such threats.

Given that phishing attacks led to approximately $4.5 billion in financial losses in 2014 alone, even a modest 1% reduction in these attacks—achieved through improved detection and prevention—could lead to a significant reduction in financial losses, potentially saving millions of dollars. Our model's ability to accurately detect and report malicious websites, therefore, contributes not only to enhancing internet security but also to reducing the economic burden on individuals and organizations alike.

With these results, our technology stands as a powerful tool in reducing the number of fake websites and, consequently, the financial impact of phishing. This positions our model as an essential component in fostering a safer online environment, boosting user confidence, and enhancing overall security. As phishing schemes continue to evolve, the continual refinement of our model will be crucial in maintaining and improving upon these protective measures.

REFERENCES

[1] Greg Aaron and Ronnie Manning. [n. d.]. APWG Phishing Reports. APWG. 2014.

[2] Iztok Fister, Dusan Fister, and Xin She Yang. 2013. A hybrid bat algorithm. Elektrotehniški Vestnik/Electrotechnical Review 80, 1-2 (2013), 1–7.

[3] François Chollet et al. 2015. Keras. https://keras.io.

[4] Federal Bureau of Investigation of USA - Internet Crime Complaint Center. 2016. Internet Crime Report. Technical Report. Federal Bureau of Investigation of USA.

[5] M. Lichman. 2013. UCI Machine Learning Repository. Available at http://archive.ics.uci.edu/ml.

[6] J. Shad and S. Sharma, "A Novel Machine Learning Approach to Detect Phishing Websites Jaypee Institute of Information Technology," pp. 425–430, 2018.

[7] Y. Sönmez, T. Tuncer, H. Gökal, and E. Avci, "Phishing web sites features classification based on extreme learning machine," 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018.

[8] T. Peng, I. Harris, and Y. Sawa, "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, 2018.

[9] M. Karabatak and T. Mustafa, "Performance comparison of classifiers on reduced phishing website dataset," 6th Int. Symp. Digit. Forensic Secur. ISDFS 2018 - Proceeding, vol. 2018–Janua, pp. 1–5, 2018

[10] S. Parekh, D. Parikh, S. Kotak, and P. S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," in 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, vol. 0, no. Icicct, pp. 949–952.

[11] Amani Alswailem, Bashayr Alabdullah, Norah Alrumayh and Aram Alsedrani,"Detecting Phishing Websites Using Machine Learning", 2019 2nd International Conference on Computer Applications &Information Security (ICCAIS).

[12] Joby James, Sandhya L. and Ciza Thomas, "Detection of Phishing URLs Using Machine Learning Techniques", 2013 International Conference on Control Communication and Computing (ICCC).

[13] Rishikesh Mahajan and Irfan Siddavatam , "Phishing Website Detection using Machine Learning Algorithms", International Journal of Computer Applications (0975 – 8887) Volume 181 – No. 23, October 2018.

[14] Jehad Ali , Rehanullah Khan , Nasir Ahmad and Imran Maqsood, "Random Forests and Decision Trees", IJCSI International Journal of Computer Science Issues, Vol. 9, Issue 5, No 3, September 2012.

[15] Gerard Biau, "Analysis of a Random Forests Model", Journal of Machine Learning Research 13(2012) 1063-1095.