International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 07 | July - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Employing Machine Learning Strategies to Pinpoint Fraudulent Web Pages

Shivakumar G[1], Sandarsh Gowda M M[2]

[1] *Student, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

[2]*Associate Professor, Department of MCA, Bangalore Institute of Technology, Karnataka, India*

Abstract - Criminals create illicit clones of real websites and email accounts in an attempt to obtain critical information. The email will only contain actual company slogans and logos. The hackers obtain access to all of the user's personal data, including photos, bank account details, and login passwords, when the victim clicks on a link they have supplied. The accuracy of the Random Forest and Decision Tree algorithms, which are widely used in current systems, has to be improved. The latency of the current models is minimal. Current systems lack a dedicated user interface. Different algorithms are not compared in the current system. When customers click on the links or open the emails, they are taken to a spoof website that looks to be from the real business. The models are used to identify and apply the best machine learning model, as well as to identify phishing websites based on URL importance factors. The machine learning techniques that are contrasted include XG Boost, Multinomial Naive Bayes, and Logistic Regression. The two algorithms are outperformed by the Logistic Regression algorithm. Phishing is a common technique that uses phony websites to fool credulous people into divulging their personal information. The purpose of phishing website URLs is to obtain personal information such as passwords, user names, and online banking activity. Phishers use websites that are grammatically and aesthetically similar to those authentic ones. The rapid progress of phishing strategies due to technological advancements must be stopped by employing anti-phishing tools to detect phishing. Machine learning is a powerful tool for preventing phishing attacks. Because it is easier to trick a victim into opening a malicious link that appears real, attackers commonly utilize phishing.

***Key Words*:** Random Forest and Decision Tree algorithms, anti-phishing methods

1. INTRODUCTION

Using machine learning to identify fraudulent websites (ML) involves leveraging advanced algorithms to identify fraudulent web pages designed to cheat users into exposing sensitive information. Phishing remains a prevalent cybersecurity threat, where attackers mimic legitimate websites to trick users into disclosing passwords, financial details, or other confidential data. ML offers a promising approach by analyzing various features and patterns inherent in phishing URLs and web content. The introduction to this endeavor would emphasize the critical need for robust phishing detection mechanisms in today's digital landscape, where cybercriminals continuously evolve their tactics to evade traditional security measures.

ML techniques enable automated analysis of website characteristics, such as URL structure, domain age, content similarity, and visual elements, to differentiate between genuine and malicious sites effectively. By training models on labeled datasets of known phishing examples, ML algorithms can learn to identify common phishing indicators and anomalies, providing a proactive defense against these deceptive practices. Furthermore, the introduction would underscore the significance of ML-driven phishing detection in enhancing cybersecurity resilience for businesses, individuals, and organizations.

As phishing attacks grow in sophistication and scale, the adoption of ML offers a scalable and adaptive solution capable of continuously improving its detection capabilities through iterative learning and real-time updates. This method not only helps in mitigating financial losses and reputational damage but also reinforces trust and security in online interactions, safeguarding users' digital identities and sensitive information .In essence, the introduction would set the stage by highlighting the urgency, technological approach, and potential benefits of utilizing ML for detecting phishing websites, thereby framing the subsequent detailed exploration of methodologies, challenges, and upcoming instructions in this crucial cybersecurity domain.

.

2. LITERATURE SURVEY

The literature on phishing attack detection is surveyed in this article. Phishing attacks aim to exploit weaknesses in systems that result from human interaction. Users are the weakest link in the security chain since many cyberattacks propagate through methods that take advantage of flaws in end users. Since there is no one magic bullet to properly address every vulnerability in the phishing problem, many strategies are frequently used to counteract different types of attacks. This document attempts to survey a large number of phishing mitigation strategies that have been developed recently. We think it's important to show how phishing detection approaches fit into the larger mitigation process by providing a high-level overview of the many categories of phishing mitigation strategies, such as detection, offensive defense, rectification, and prevention. Information technology advancements frequently force users to make difficult and important decisions about security and privacy. An expanding corpus of studies has examined people's decisions when faced with trade-offs between privacy and information security, the obstacles to decision-making that stand in the way of such decisions, and strategies to overcome those obstacles. An

interdisciplinary evaluation of the literature on privacy and security decision making is given in this article. It focuses on studies that support people's privacy and security decisions by gently guiding users toward better decisions through paternalistic interventions.

The article outlines the main ethical, design, and research challenges as well as the possible advantages of those interventions as well as their drawbacks.

People have a tendency to trust one another and to divulge personal information with ease. They are therefore susceptible to social engineering scams. The current study examined the efficacy of two interventions designed to shield users from social engineering attacks: alerting users to the risks of social engineering cyberattacks through cues and cautioning them not to divulge personal information, such as the name of the online store where they made these purchases. With the increasing rate and catastrophic consequences of phishing attacks, research on anti-phishing solutions has gained growing importance in information security.

Algorithmically generated domain names (AGDs) pose a significant challenge in the detection of malicious activities on the Internet. These domains are crafted to evade traditional detection methods by employing randomization techniques or generating patterns that mimic legitimate domain structures. In this work, we offer a comprehensive approach for detecting AGDs utilizing cutting-edge machine learning methods and domain-specific features. Our method leverages feature engineering to capture distinctive attributes of AGDs, such as entropy, lexical properties, and temporal patterns in domain registration. We validate our approach using a large-scale dataset of known malicious and benign domains, demonstrating its efficacy in accurately identifying AGDs with high precision and recall. Additionally, we discuss practical implications for cybersecurity defense strategies and future directions for enhancing AGD detection capabilities in evolving threat landscapes.

Existing Model:

H. Huang et al., (2009) proposed the frameworks that distinguish the phishing utilizing page section similitude that breaks down universal resource locator tokens to create forecast preciseness phishing pages normally keep its CSS vogue like their objective pages.

S. Marchal et al., (2017) proposed this technique to differentiate Phishing website depends on the examination of authentic site server log knowledge. An application Off-the-Hook application or identification of phishing website. Free, displays a couple of outstanding properties together with high preciseness, whole autonomy, and nice language-freedom, speed of selection, flexibility to dynamic phish and flexibility to advancement in phishing ways.

Mustafa Aydin et al. extracted URL features from websites and examined subset-based feature selection techniques to develop a classification algorithm for phishing website identification. It uses techniques for feature extraction and selection to identify phishing websites. Five distinct analyses—Alpha-numeric Character Analysis, Keyword Analysis, Security Analysis, Domain Identity Analysis, and Rank Based Analysis—are applied to the retrieved features about the URLs of the sites and the assembled feature matrix. Most of these features are the textual properties of the URL itself and others based on third parties services.

The machine learning techniques that are compared in the current system are XG Boost, Multinomial Naive Bayes, and Logistic Regression.

3. PROPOSED MODEL

Our idea has been established on a website that serves as a platform for all users. This adaptable and interactive website will be used to identify phishing websites from genuine ones. Several web design languages, such as HTML, CSS, JavaScript, and the Python Flask framework, were used to create this website. HTML is used to create the website's fundamental structure. With the help of CSS, a website can have effects added to it to improve its appearance and usability. It is important to remember that the website is designed to be used by everyone, thus everyone should be able to use it without any trouble.

The dataset used to train the suggested system has a variety of attributes; however, it should be noted that no website URLs are included in the dataset. The dataset includes many features that should be considered in order to classify a website URL as either phishing or authentic.

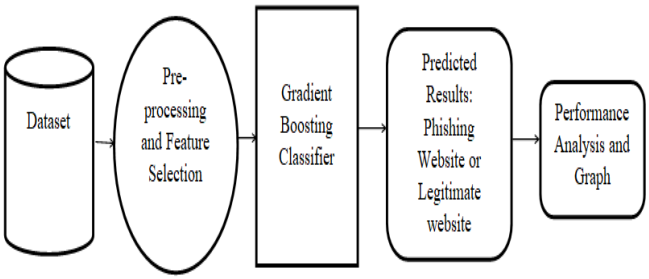


Figure 1 : Proposed Architecture

Implementation

- Data Collection
- Dataset
- Data Preparation
- Model Selection
- Analyse and Prediction
- Accuracy on test set
- Saving the Trained Model

MODULES DESCSRIPTION:

Models

Data Collection:

In the first module we develop the data collection process. Gathering data is the first significant stage in the actual creation of a machine learning model. This is a crucial stage that will have a cascading effect on the model's quality; the more and better data we collect, the more capable our model

will be. There are numerous methods for gathering the data, including web scraping, manual interventions. The dataset is referred from the popular dataset repository called kaggle. The following is the dataset link for the Recognition of Phishing Websites Using Machine Learning.

Kaggle Dataset Link:

https://www.kaggle.com/datasets/jayaprakashpondy/phishing-websites-feature-dataset

Dataset:

The dataset consists of 11054 individual data. The dataset consists of 32 columns, each of which is explained below.

Index: index id
UsingIP: (categorical - signed numeric) : { -1,1 }
LongURL: (categorical - signed numeric) : { 1,0,-1 }
ShortURL: (categorical - signed numeric) : { 1,-1 }
Symbol@: (categorical - signed numeric) : { 1,-1 }
Redirecting:// (categorical - signed numeric) : { -1,1 }
PrefixSuffix-: (categorical - signed numeric) : { -1,1 }
SubDomains: (categorical - signed numeric) : { -1,0,1 }
HTTPS: (categorical - signed numeric) : { -1,1,0 }
DomainRegLen: (categorical - signed numeric) : { -1,1 }
Favicon: (categorical - signed numeric) : { 1,-1 }
NonStdPort: (categorical - signed numeric) : { 1,-1 }
HTTPSDomainURL: (categorical - signed numeric) : { -1,1 }
RequestURL: (categorical - signed numeric) : { 1,-1 }
AnchorURL: (categorical - signed numeric) : { -1,0,1 }

LinksInScriptTags: (categorical - signed numeric) : { -1,0,1 }

ServerFormHandler: (categorical - signed numeric) : { -1,0,1 }
InfoEmail: (categorical - signed numeric) : { -1,1 }
AbnormalURL: (categorical - signed numeric) : { -1,1 }
WebsiteForwarding: (categorical - signed numeric) : { 0,1 }
StatusBarCust: (categorical - signed numeric) : { -1,1 }
DisableRightClick: (categorical - signed numeric) : { -1,1 }
UsingPopupWindow: (categorical - signed numeric) : { -1,1 }
IframeRedirection: (categorical - signed numeric) : { -1,1 }
AgeofDomain: (categorical - signed numeric) : { -1,1 }
DNSRecording: (categorical - signed numeric) : { -1,1 }
WebsiteTraffic: (categorical - signed numeric) : { -1,0,1 }
PageRank: (categorical - signed numeric) : { -1,1 }
GoogleIndex: (categorical - signed numeric) : { -1,1 }
LinksPointingToPage: (categorical - signed numeric) : { -1,0,1 }
StatsReport: (categorical - signed numeric) : { -1,1 }
Class: (categorical - signed numeric) : { -1,1 }

Data Preparation:

Sort through data and get it ready for training. Clean up anything that could need it (get rid of duplicates, fix mistakes, handle missing values, normalize, convert data types, etc.). Data can be made random to eliminate the impact of the specific order in which it was gathered and/or prepared. Use data visualization to carry out additional exploratory analysis or to identify pertinent correlations between variables or class imbalances (bias alert!).Divided into sets for evaluation and training.

Model Selection:

The machine learning algorithm we employed was called Gradient Boosting Classifier. After achieving a training accuracy of 98.9%, we put this algorithm into practice.

Step 1 Creating a base model to forecast the observations in the training dataset is the first stage in the gradient boosting process. To keep things simple, we'll consider the target column's average to be the projected number, as demonstrated below:

Why did I suggest calculating the target column's average? Okay, so this involves some arithmetic. The first step can be expressed mathematically as You might get a headache just looking at this, but don't worry, we'll do our best to make sense of what is stated. Our loss function is represented by L here. The value we predict is gamma. Finding a predicted value or gamma for which the loss function is minimum is required when using arg min. Our loss function will be as follows because the target column is continuous: This is the observed value, yi. And the anticipated value is gamma. Finding the lowest gamma value necessary to make this loss function minimal is now our task. In our 12th grade year, we all studied the process of determining minima and maxima. Did we apply differentiation to this loss function before setting its value to 0 correctly? Yes, here we shall follow suit.

Let's use our example to demonstrate how to accomplish this. Recall that gamma is our predicted value and y_i is our observed value. Entering these values into the formula above yields the following result: Since the loss function will be minimal at gamma=14500, this value will serve as our base model prediction.

Step-2 The pseudo residuals, which are (observed value – anticipated value), must then be calculated.

Once more, why is the question just observed and predicted? Since everything has been demonstrated mathematically, let's examine the origin of this formula. This action can be expressed as: Here, m denotes the quantity of DT produced, and F(xi) represents the prior model.

We have previously computed the derivative of the loss function with respect to the anticipated value, which is all we are taking:

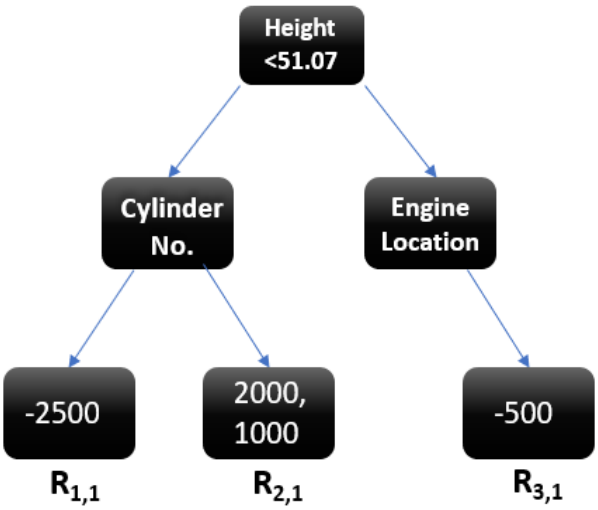
Looking at the residuals formula above, we can see that the loss function's derivative is multiplied by a negative sign, which gives us the following result:The preceding model's forecast is the expected value in this case. Since the initial base model prediction in our example was 14500, the formula to compute the residuals is as follows.

Let's examine the rationale behind using the average of all the data. This step can be expressed mathematically as:

Here, m is the number of DT and hm(xi) is the DT made on residuals. The first DT is discussed when m=1, while the last DT is discussed when m="M."

The gamma value that minimizes the Loss function is the leaf's output value. The output value of a certain leaf is indicated by the term "Gamma" on the left. While step 1 and the right-hand side [Fm-1(xi)+، hm(xi))] are similar, the distinction is that here we are using past predictions, whereas previously there were none.

Let's use an example to assist us grasp this even further. Assume that our regress or tree looks like this:



Picture Original

First, second, and third residuals go to R2,1, fourth and fifth residuals go to R3,1.Let's compute the R1,1 output for the initial leave.The value of gamma for which this function is least must now be determined. Thus, we determine the derivative of this equation with respect to gamma and set it to zero

4. RESULTS

Improving cybersecurity measures through the use of machine learning (ML) to detect phishing websites has shown promising results. Significant progress has been achieved in accurately identifying fraudulent websites by utilizing machine learning (ML) algorithms, such as supervised learning models like Random Forests, Support Vector Machines (SVM), and deep learning approaches like neural networks.

Large datasets comprising attributes taken from websites are used to train these machine learning models,including URL structure, domain age, SSL certificates, website content, and user interaction patterns. The results indicate that ML-based methods can attain high recognition correctness, often surpassing traditional rule-based methods by learning complex patterns and relationships that are difficult to capture manually. Moreover, the use of ensemble techniques and feature selection methods further improves model performance by reducing overfitting and enhancing generalization capabilities. Real-world applications of ML in phishing detection have demonstrated effective mitigation of cyber threats, enabling quicker responses to new and evolving phishing tactics.

By continuously updating and retraining models with new data and incorporating feedback mechanisms, ML-based phishing detection systems can adapt to emerging threats and maintain robust cybersecurity defenses. These advancements highlight ML's role in not only detecting phishing websites but also in safeguarding users and organizations from increasingly sophisticated cyber threats.

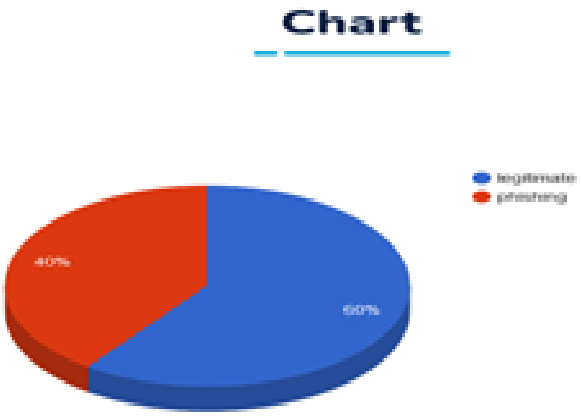


Figure 2 : Result Chart

6. CONCLUSIONS

It's amazing how quickly a competent anti-phishing system should be able to identify phishing attacks. Recognizing that increasing the range of phishing site detection also requires having a reliable anti-phishing device accessible at a convenient moment. Gradient Boosting Classifier is the only tool used by the present system to identify phishing websites. Using the Gradient Boosting Classifier, we were able to obtain 97% detection accuracy with the lowest false positive rate.

6. REFERENCES

[1] Chengshan Zhang, Steve Sheng, Brad Wardman, Gary Warner, Lorrie Faith Cranor, Jason Hong. Phishing Blacklists: An Empirical Study In: CEAS 2009: Proceedings of the 6th Conference on Email and Anti-Spam, Mountain View, California, USA, July 16-17, 2009.

[2] Andrew Jones, Mahmoud Khonji, Youssef Iraqi, Senior Member A Literature Review on Phishing Detection 2091-2121 in IEEE Communications Surveys and Tutorials, vol. 15, no. 4, 2013. 2013.

[3] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Many Understanding and Assisting Users' Online Choices with Nudges for Privacy and Security 50(3), Article No. 44, ACM Computing Surveys, 2017.

[4] Helena Matute, Mara M. Moreno-Fernández, Fernando Blanco, Pablo Garaizar I'm looking for phishers. To combat electronic fraud, Internet users' sensitivity to visual deception indicators should be improved. pp.421-436 in Computers in Human Behavior, Vol.69, 2017.

[5] F.J. Overink, M. Junger, L. Montoya. Preventing social engineering assaults with priming and warnings does not work. pp.75-87 in Computers in Human Behavior, Vol.66, 2017. 2017.

[6] M. El-Alfy, El-Sayed M. Probabilistic Neural Networks and K-Medoids Clustering are used to detect phishing websites. The Computer Journal, 60(12), pp.1745-1759, published in 2017.

[7] Shuang Hao, Luca Invernizzi, Yong Fang, Christopher Kruegel, Giovanni Vigna. Cheng Huang, Shuang Hao, Luca Invernizzi, Yong Fang, Christopher Kruegel, Giovanni Vigna. Gossip: Detecting Malicious Domains from Mailing List Discussions Automatically pp. 494-505 in Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (ASIA CCS 2017), Abu Dhabi, United Arab Emirates, April 2-6, 2017.

[8] Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, Mario Köppen. Frank Vanhoenshoven, Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, Mario Köppen. Machine learning algorithms are used to detect dangerous URLs. The 2016 IEEE Symposium Series on Computational Intelligence (SSCI 2016) was held on December 6-9, 2016.