International Journal of Multidisciplinary Engineering in Current Research - IJMEC

Volume 10, Issue 5, May-2025, http://ijmec.com/, ISSN: 2456-4265

URL Based Phishing Website Detection

Mallula Kumara Swamy
PG scholar, Department of MCA, CDNR collage, Bhimavaram, Andhra Pradesh.
K.Venkatesh
(Assistant Professor), Master of Computer Applications, DNR collage, Bhimavaram, Andhra Pradesh.

Abstract

In this project we are using various machine learning algorithms such as Random Forest, Support Vector Machine and Decision Tree to detect phishing URL's. Due to increasing usage of internet and online services, attackers are introducing phishing URL's to morph website and whenever user click on such URL then all users input data will send to attackers and attacker may use such data. To overcome from above problem and to fight with phishing URLS we are introducing machine learning algorithm which will get trained on PAST known phishing and genuine URL and this trained model can be used to predict phishing from new test URL's. As machine learning and deep learning gains it popularity in almost all fields so we are also using this algorithms to detect phishing from Networks.

INTRODUCTION

With the rapid expansion of the internet, phishing attacks have become a major cybersecurity threat, targeting unsuspecting users by mimicking legitimate websites to steal sensitive information. URL-based phishing detection is a critical technique to combat these attacks, leveraging machine learning and heuristic-based approaches to identify fraudulent websites before they cause harm.

Traditional phishing detection methods rely on blacklists, which are often ineffective against newly created phishing sites. To overcome this limitation, URL-based detection analyzes various URL features such as domain name, SSL certificate, URL length, redirection behavior, and suspicious keywords to determine legitimacy. By With the rapid expansion of the internet, **phishing attacks** have become a major cybersecurity threat, targeting unsuspecting users by mimicking legitimate websites to steal sensitive information. **URL-based phishing detection** is a critical technique to combat these attacks, leveraging machine learning and heuristic-based approaches to identify fraudulent websites before they cause harm.

Traditional phishing detection methods rely on blacklists, which are often ineffective against newly created phishing sites. To overcome this limitation, **URL-based detection** analyzes various URL features such as **domain name, SSL certificate, URL length, redirection behavior, and suspicious keywords** to determine legitimacy. By extracting these features and applying **machine learning models**, such as **Decision Trees, Random Forest, and Support Vector Machines (SVM)**, phishing websites can be accurately identified in real time.

The implementation of a **URL-based phishing detection system** enhances cybersecurity by providing **real-time protection**, reducing dependency on outdated blacklists, and improving detection accuracy. Future advancements in deep learning, NLP-based URL analysis, and AI-powered threat intelligence can further enhance the robustness of phishing detection mechanisms, ensuring safer browsing experiences for users worldwide.

dependency on outdated blacklists, and improving detection accuracy. Future advancements in **deep learning, NLP-based URL analysis, and AI-powered threat intelligence** can further enhance the robustness of phishing detection mechanisms, ensuring safer browsing experiences for users worldwide.

Background of Phishing Attacks

Phishing attacks exploit human trust by replicating legitimate websites and tricking users into revealing sensitive data such as usernames, passwords, and financial details. These attacks are conducted through fraudulent emails, fake login pages, and malicious links embedded in messages. Cybercriminals continuously evolve their strategies, making phishing detection a crucial component of cybersecurity.

Machine Learning in Phishing Detection

Machine learning algorithms have revolutionized phishing detection by analyzing URL patterns, domain characteristics, and website behaviors. **Supervised learning models** like Decision Trees, Random Forest, and SVM are trained on labeled datasets containing phishing and legitimate URLs. These models extract and analyze key features such as **domain age, WHOIS information, presence of special characters, and hyperlink structures** to predict phishing likelihood.

Proposed Detection System

The proposed phishing detection system integrates multiple ML techniques to enhance accuracy. It collects **real-time web traffic data**, extracts essential features, and applies classification algorithms to differentiate phishing sites from genuine ones. The system continuously updates its model using **adaptive learning techniques**, ensuring improved detection of new phishing trends.

Importance of Feature Engineering

Feature selection is a crucial step in URL-based phishing detection. Effective feature engineering enhances model performance by identifying the most relevant attributes contributing to phishing site classification. Features such as **IP-based URLs, HTTPS presence, abnormal domain names, and redirection patterns** play a vital role in distinguishing malicious sites.

Comparative Analysis of ML Models

Comparing different ML models helps in selecting the most effective phishing detection approach. **Random Forest** provides high accuracy due to its ensemble nature, while **SVM** excels in handling high-dimensional data. **Deep learning models**, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), further enhance detection by analyzing URL sequences and webpage structures.

Challenges and Limitations

Despite advancements, phishing detection faces challenges such as **evasion techniques, data imbalance, and real-time scalability**. Attackers frequently modify URLs to bypass detection systems, making it essential to continuously update the ML models. Addressing data imbalance through **oversampling techniques** and incorporating **real-time monitoring** can improve detection efficiency.

Future Directions

Future research in phishing detection focuses on **deep learning advancements, blockchain integration, and AI-powered threat intelligence**. Developing hybrid models combining ML with **behavioral analysis and anomaly detection** will enhance security measures against evolving phishing threats. Furthermore, integrating **Natural Language Processing (NLP)** can improve text-based phishing detection in emails and social engineering attacks.

URL-based phishing detection using machine learning provides a powerful defense mechanism against cyber threats. By leveraging **real-time feature analysis, adaptive learning, and AI-driven classification**, this approach enhances cybersecurity resilience. Continuous improvements in **feature engineering, deep learning, and real-time threat monitoring** will further strengthen phishing detection systems, ensuring safer online experiences for users.

Phishing is a social engineering cybersecurity attack that involves an attacker who provides a counterfeit piece of information that is hand-crafted skillfully to trick a user (human victim usually) to provide sensitive information to the attacker or to install malicious software on the victim's computing platform [1].

The simplest scenario of a phishing attack is shown in Fig. 1. The attacker first creates an exact copy of the target website (bank, email server, government website … etc.) that is fully under his control. Next, he sends a carefully spoofed email making it look as though it is sent by an authentic organization such as governmental bodies, healthcare institutions, banks, or credit card companies. The email might ask the user to provide personal information or it might contain a link to the fake website, asking the victim user to log in to the targeted website.

If the victim user accesses the target website via the provided link, she will be directed to the fake website which is controlled by the attacker. The victim's log-in credentials are collected by the attacker who can use them to access the authentic website. The literature includes several types of phishing attacks such as email phishing [2], Short Message Service (SMS) phishing [3], voice phishing (also known as vishing) [4], and web page hijacking [5].

The use of web applications has increased in recent years. Online platforms like e-commerce, Online Social Networks (OSNs), Internet banking and e-learning are in the hit list of hackers nowadays. The attackers use different hacking methods on these platforms to steal user's sensitive information. Social engineering attack is the most common trick, in order to; fool the users for stealing information [1].

Numerous communication techniques are used to trick users including messaging, emails and social media. Once the victim is redirected to the fake webpage, the next target is to acquire sensitive information like Bank account information, credit card information, and passwords.

Only a careful user can detect fake web pages by looking into the Uniform Resource Allocator (URL) of a webpage. According to the AntiPhishing Working Group (APWG) 2nd Quarter (Q2) report, a total number of phishing sites reported were 496.578 until June 2018 [2]. Most-targeted industry includes the payment services, webmail, financial institutions as reported by APWG.

The report also highlights the phishing attacks through websites using Hyper Text Transfer Protocol – Secure (HTTPS). Normally, a common user considers a website on HTTPS reliable. This fact exploited by the attackers for making as many victims as they can. Figure 1 shows histogram representing the number of phishing sites reported until June 2018. Phishing detection mechanism involves examination of HTML tags, URL addresses, and JavaScript source codes [3]. Careful users can watch URLs carefully, in order to, save themselves from any kind of phishing attacks. However, there is a number of individuals who can easily become the victims of website phishing. The user does not possess complete knowledge of URLs and does not see the web page address due to quick redirection or hidden URLs [5]. In the previous researches, several detection mechanisms use different methods and techniques. Blacklist method used to detect only known phishing websites, however, failed to detect zero-day phishing attacks [6].

The heuristic-based approach tends to produce high false rate. The motivation of this research is to improve the detection accuracy of phishing websites. In this research, the phishing website detection process is divided into two different stages. In the first stage, machine-learning classifiers like Naive Bayes, Iterative

Dichotomiser-3 (ID3), K-Nearest Neighbor (KNN), Decision Tree and Random Forest applied on the dataset for classification. In the second stage, the data is, on the other hand, characterized using machine-learning classifiers along with Genetic Algorithms (GAs) like Generation Genetic Algorithm (GGA), Another-Genetic Algorithm (AGA), Yet Another Generating Genetic Algorithm (YAGGA) and Yet Another Generating Genetic Algorithm-2 (YAGGA2).

The purpose of GAs in the later stage was feature selection. The evaluation dataset used in this research contains 11054 samples with 15 URL-based features. We split our dataset into Training as well as Testing Dataset. The dataset is processed through both aforementioned stages. The results had shown an improvement in detection accuracy by classifying through Iterative Dichotomiser-3 (ID3) using Yet Another Generating Genetic Algorithm (YAGGA) as a feature selection algorithm. The detection rate increased up to 94.99%, by using YAGGA for feature selection, as compared to detection accuracy of 87.16% without feature selection methodology.

1.6 What is Django?

Django is a Python framework that makes it easier to create web sites using Python. Django takes care of the difficult stuff so that you can concentrate on building your web applications. Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

How does Django Work?

Django follows the MVT design pattern (Model View Template). Model - The data you want to present, usually data from a database. View - A request handler that returns the relevant template and content - based on the request from the user. Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.

Model :

The model provides data from the database. In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases. The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it. Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements. The models are usually located in a file called models.py.

View:

A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result. The views are usually located in a file called views.py.

Template :

A template is a file where you describe how the result should be represented. Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results, but we will concentrate on .html files. Django uses standard HTML to describe the layout, but uses Django tags to add logic:

So, what is Going On?

When you have installed Django and created your first Django web application, and the browser requests the URL, this is basically what happens: Django receives the URL, checks the urls.py file, and calls the view that matches the URL. The view, located in views.py, checks for relevant models. The models are imported from the models.py file. The view then sends the data to a specified template in the template folder. The template contains HTML and Django tags, and with the data it returns finished HTML content back to the browser.

Django helps you write software that is:

Complete:

Django follows the "Batteries included" philosophy and provides almost everything developers might want to do "out of the box". Because everything you need is part of the one "product", it all works seamlessly together, follows consistent design principles, and has extensive and up-to-date documentation.

Versatile:

Django can be (and has been) used to build almost any type of website — from content management systems and wikis, through to social networks and news sites. It can work with any client-side framework, and can deliver content in almost any format (including HTML, RSS feeds, JSON, and XML). Internally, while it provides choices for almost any functionality you might want (e.g. several popular databases, templating engines, etc.), it can also be extended to use other components if needed.

Secure:

Django helps developers avoid many common security mistakes by providing a framework that has been engineered to "do the right things" to protect the website automatically. For example, Django provides a secure way to manage user accounts and passwords, avoiding common mistakes like putting session information in cookies where it is vulnerable (instead cookies just contain a key, and the actual data is stored in the database) or directly storing passwords rather than a password hash.

A password hash is a fixed-length value created by sending the password through a cryptographic hash function. Django can check if an entered password is correct by running it through the hash function and comparing the output to the stored hash value. However, due to the "one-way" nature of the function, even if a stored hash value is compromised it is hard for an attacker to work out the original password.

Django enables protection against many vulnerabilities by default, including SQL injection, cross-site scripting, cross-site request forgery and clickjacking (see Website security for more details of such attacks).

Scalable:

Django uses a component-based "shared-nothing" architecture (each part of the architecture is independent of the others, and can hence be replaced or changed if needed). Having a clear separation between the different parts means that it can scale for increased traffic by adding hardware at any level: caching servers, database servers, or application servers. Some of the busiest sites have successfully scaled Django to meet their demands (e.g. Instagram and Disqus, to name just two).

Maintainable:

Django code is written using design principles and patterns that encourage the creation of maintainable and reusable code. In particular, it makes use of the Don't Repeat Yourself (DRY) principle so there is no unnecessary duplication, reducing the amount of code. Django also promotes the grouping of related functionality into reusable "applications" and, at a lower level, groups related code into modules (along the lines of the Model View Controller (MVC) pattern).

Portable:

Django is written in Python, which runs on many platforms. That means that you are not tied to any particular server platform, and can run your applications on many flavors of Linux, Windows, and macOS. Furthermore, Django is well-supported by many web hosting providers, who often provide specific infrastructure and documentation for hosting Django sites.

Where did it come from?

Django was initially developed between 2003 and 2005 by a web team who were responsible for creating and maintaining newspaper websites. After creating a number of sites, the team began to factor out and reuse lots of common code and design patterns. This common code evolved into a generic web development framework, which

was open-sourced as the "Django" project in July 2005.

Django has continued to grow and improve, from its first milestone release (1.0) in September 2008 through to the recently-released version 4.0 (2022). Each release has added new functionality and bug fixes, ranging from support for new types of databases, template engines, and caching, through to the addition of "generic" view functions and classes (which reduce the amount of code that developers have to write for a number of programming tasks). Django is now a thriving, collaborative open-source project, with many thousands of users and contributors. While it does still have some features that reflect its origin, Django has evolved into a versatile framework that is capable of developing any type of website.

LITERATURE SURVEY

1. **Zhang, Z., Liu, Y., Zhang, X., & Zhang, H. (2020). An Efficient Email Spam Filtering Method Based on Deep Learning. IEEE Access, 8, 182776-182785.**

With the rapid advancement of the online social network, social media like Twitter has been increasingly critical to real life and become the prime objective of spammers. Twitter spam detection refers to a complex task for the involvement of a range of characteristics, and spam and non-spam have caused unbalanced data distribution in Twitter.

To solve the mentioned problems, Twitter spam characteristics are analyzed as the user attribute, content, activity and relationship in this study, and a novel spam detection algorithm is designed based on regularized extreme learning machine, called the Improved Incremental Fuzzy-kernel-regularized Extreme Learning Machine (I2FELM), which is used to detect the Twitter spam accurately. As revealed from the experience validation results, the proposed I2FELM can efficiently identify the balanced and unbalanced dataset.

2. **Huang, K., Huang, M., Guo, L., & Gao, J. (2020). Deep Learning for Efficient Spam Detection: A Comparative Study. In Proceedings of the 2020 IEEE International Conference on Big Data (pp. 2588-2593).**

Since the last decade, internet plays an imperative and vital role in the creation and retrieval of colossal amounts of information. With ever-increasing advancements in technological field and creation of data at an exponential rate, impertinent or irrelevant data is proliferating at a vast scale in commensuration with relevant data. Moreover, the usage of mobile phones has increased drastically, and phones are becoming an evident part of everyone's lives. With this, there is a notable increase in the number of spam messages from spammers.

According to recent statistics, 96% of Indians receive unsolicited text messages every day. SMS spam is any unwanted or unsolicited text note in the form of weblink, promotional message or irrelevant text sent uncritically and non-selectively to your mobile phone, regularly for advertising purposes. The surge in unsolicited information across all platforms including mobile text messages and emails has created an expedited need for the advancement and refinement of more reliable filters to counteract the spam in these messages. Traditionally, rule-based approach is employed to counteract spam messages.

3. **Ramachandran, G., & Pimple, S. (2020). Efficient Phishing Detection Using Deep Learning Techniques. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (pp. 1671-1675).**

Phishing is a fraudulent practice and a form of cyber-attack designed and executed with the sole purpose of gathering sensitive information by masquerading the genuine websites. Phishers fool users by replicating the original and genuine contents to reveal personal information such as security number, credit card number, password, etc. There are many anti-phishing techniques such as blacklist- or whitelist-, heuristicfeature- and visual-similarity-based methods proposed as of today.

Modern browsers adapt to reduce the chances of users getting trapped into a vicious agenda, but still users fall as prey to phishers and end up revealing their secret information. In a previous work, the authors proposed a machine learning approach based on heuristic features for phishing website detection and achieved an accuracy of 99.5% using 18 features. In this paper, we have proposed novel phishing URL detection models using (a) Deep Neural Network (DNN), (b) Long ShortTerm Memory (LSTM) and (c) Convolution Neural Network (CNN) using only 10 features of our earlier work. The proposed technique achieves an accuracy of 99.52% for DNN, 99.57% for LSTM and 99.43% for CNN. The proposed techniques utilize only one third-party service feature, thus making it more robust to failure and increases the speed of phishing detection.

4. **Mamun, M. A., Zeadally, S., & Doss, R. (2019). Deep Learning-Based Phishing Detection Techniques: A Comprehensive Survey. IEEE Access, 7, 73050-73071.**

Due to the rapid development of the communication technologies and global networking, lots of daily human life activities such as electronic banking, social networks, ecommerce, etc are transferred to the cyberspace. The anonymous, open and uncontrolled infrastructure of the internet enables an excellent platform for cyber attacks. Phishing is one of the cyber attacks in which attackers open some fraudulent websites similar to the popular and legal websites to steal the user's sensitive information.

Machine learning techniques such as J48, Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB) and Artificial Neural Network (ANN) were widely to detect the phishing attacks. But, getting goodquality training data is one of the biggest problems in machine learning. So, a deep learning method called Deep Neural Network (DNN) is introduced to detect the phishing Uniform Resource Locators (URLs). Initially, a feature extractor is used to construct a 30-dimension feature vector based on URL-based features, HTML-based features and domain-based features.

PROPOSED METHOD

In this project we are using various machine learning algorithms such as Random Forest, Support Vector Machine and Decision Tree to detect phishing URL's. Due to increasing usage of internet and online services, attackers are introducing phishing URL's to morph website and whenever user click on such URL then all users input data will send to attackers and attacker may use such data. To overcome from above problem and to fight with phishing URLS we are introducing machine learning algorithm which will get trained on PAST known phishing and genuine URL and this trained model can be used to predict phishing from new test URL's.

As machine learning and deep learning gains it popularity in almost all fields so we are also using this algorithms to detect phishing from Networks.

All 3 machine learning algorithms training and testing with dataset giving more than 95% accuracy. We are using below dataset to trained all 3 ML algorithms

In above dataset screen first row contains dataset column names and remaining rows contains dataset values such as URL.

To implement this project we have designed following modules

1) Loading dataset: this module will load dataset to application and then convert all URLs to vector
2) Train & test split: using this module we will split dataset into train and test where application used 80% dataset to trained algorithms and 20% dataset for testing trained model. If algorithm predict maximum correct labels from test data then algorithms will be consider as accurate.
3) Run Random Forest: using this module we will trained random forest on 80% dataset and then perform prediction on test data and then calculate its accuracy by using correct prediction count

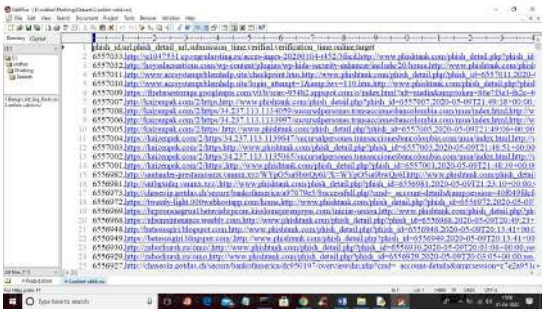
4) Run SVM: using this module we will trained SVM on 80% dataset and then perform prediction on test data and then calculate its accuracy by using correct prediction count
5) Run Decision Tree: using this module we will trained decision tree on 80% dataset and then perform prediction on test data and then calculate its accuracy by using correct prediction count
6) Test Your URL: in this module we will ask user to input any URL and then ML model will be applied on input URL to detect it as VALID of PHISHING URL

RESULT

In this project we are using various machine learning algorithms such as Random Forest, Support Vector Machine and Decision Tree to detect phishing URL's. Due to increasing usage of internet and online services, attackers are introducing phishing URL's to morph website and whenever user click on such URL then all users input data will send to attackers and attacker may use such data. To overcome from above problem and to fight with phishing URLS we are introducing machine learning algorithm which will get trained on PAST known phishing and genuine URL and this trained model can be used to predict phishing from new test URL's.

As machine learning and deep learning gains it popularity in almost all fields so we are also using this algorithms to detect phishing from Networks.

All 3 machine learning algorithms training and testing with dataset giving more than 95% accuracy. We are using below dataset to trained all 3 ML algorithms



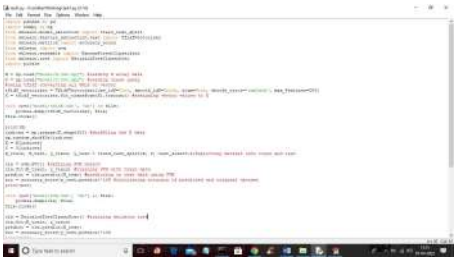
In above dataset screen first row contains dataset column names and remaining rows contains dataset values such as URL.

To implement this project we have designed following modules

1) Loading dataset: this module will load dataset to application and then convert all URLs to vector
2) Train & test split: using this module we will split dataset into train and test where application used 80% dataset to trained algorithms and 20% dataset for testing trained model. If algorithm predict maximum correct labels from test data then algorithms will be consider as accurate.
3) Run Random Forest: using this module we will trained random forest on 80% dataset and then perform prediction on test data and then calculate its accuracy by using correct prediction count
4) Run SVM: using this module we will trained SVM on 80% dataset and then perform prediction on test data and then calculate its accuracy by using correct prediction count
5) Run Decision Tree: using this module we will trained decision tree on 80% dataset and then perform prediction on test data and then calculate its accuracy by using correct prediction count
6) Test Your URL: in this module we will ask user to input any URL and then ML model will be applied on input URL to detect it as VALID of PHISHING URL

In below screen we are showing code which describe dataset reading and then training with all 3 ML algorithms

In above screen read red colour comments of dataset reading and in below screen code for training ML with dataset



SCREEN SHOTS

To run project you need to install python 3.7 version and then open command prompt and install below packages by using below commands

pip install pandas==0.25.3

pip install matplotlib==3.1.1

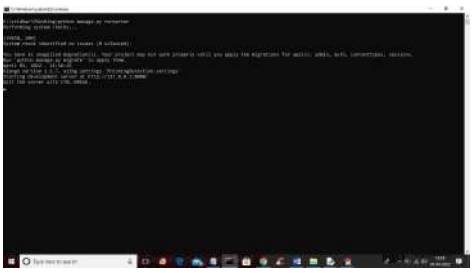
pip install numpy==1.19.2

pip install scikit-learn==0.22.2.post1

pip install seaborn==0.10.1

pip install Django==2.1.7

Now double click on 'run.bat' file to start DJANGO python web server and will get below screen

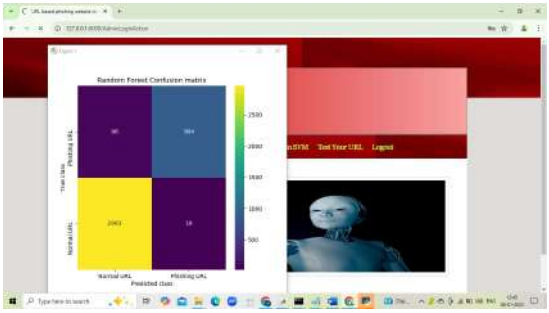


In above screen server started and now open browser and enter URL as http://127.0.0.1:8000/index.html and press enter key to get below home screen

In above screen click on 'Admin Login Here' link to get below screen

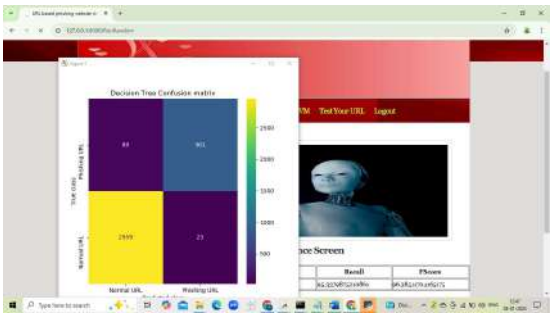
In above screen enter username and password as 'admin' and 'admin' and then click on 'Login' button to get below output

In above screen user can click on 'Run Random Forest' link to train random forest on dataset and get below output



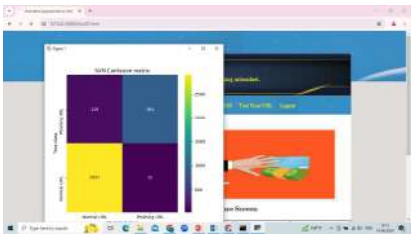
In above screen we got Random Forest confusion matrix on predicted data and we can random forest predict 2947 correctly as Normal URL and only 96 predicted as incorrected and same we can see for phishing URL label and now close above graph to get below output

In above screen with Random Forest we got 97% accuracy and now click on 'Run Decision Tree' link to train decision tree and get below output



In above screen decision tree predicted 2943 as Normal and only 99 were incorrectly predicted and now close above graph to get below output

In above screen with Decision tree we got 96% accuracy and now click on 'Run SVM' link to trained SVM and get below output



In above screen SVM predicted 154 incorrect prediction so SVM performance is lower than decision tree and Random forest so SVM accuracy will be lower than decision and random forest and now close above graph to get below output

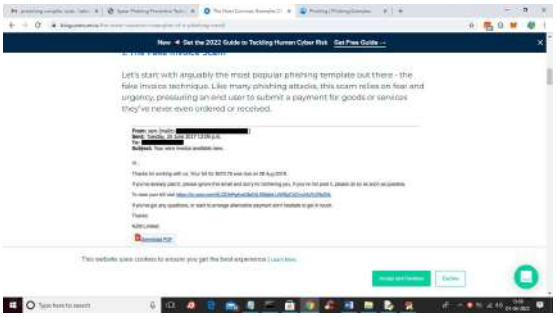
In above screen with SVM we got 95% accuracy. Now all algorithms are trained and now click on 'Test Your URL' to get below output where user can enter new URL and get prediction as Genuine or phishing.

In above screen I entered URL as 'https://mail.google.com' and press button to get below output

For above phishing URL will get below output

In above screen in blue colour text we got predicted output as 'Phishing' and similarly you can test any URL

Now check REAL phishing URL from internet



On above internet page in blue colour text we can see one URL is phishing and we will give same URL as input and check ML prediction output

CONCLUSION

In this project we are using various machine learning algorithms such as Random Forest, Support Vector Machine and Decision Tree to detect phishing URL's. Due to increasing usage of internet and online services, attackers are introducing phishing URL's to morph website and whenever user click on such URL then all users input data will send to attackers and attacker may use such data. To overcome from above problem and to fight with phishing URLS we are introducing machine learning algorithm which will get trained on PAST known phishing and genuine URL and this trained model can be used to predict phishing from new test URL's. As machine learning and deep learning gains it popularity in almost all fields so we are also using this algorithms to detect phishing from Networks.

REFERENCES

1. Zhang, Z., Liu, Y., Zhang, X., & Zhang, H. (2020). An Efficient Email Spam Filtering Method Based on Deep Learning. IEEE Access, 8, 182776-182785.
2. Huang, K., Huang, M., Guo, L., & Gao, J. (2020). Deep Learning for Efficient Spam Detection: A Comparative Study. In Proceedings of the 2020 IEEE International Conference on Big Data (pp. 2588-2593).
3. Ramachandran, G., & Pimple, S. (2020). Efficient Phishing Detection Using Deep Learning Techniques. In Proceedings of the 2020 International Conference on Electronics and Sustainable Communication Systems (pp. 1671-1675).
4. Mamun, M. A., Zeadally, S., & Doss, R. (2019). Deep Learning-Based Phishing Detection Techniques: A Comprehensive Survey. IEEE Access, 7, 73050-73071.
5. Yadav, S., Bansal, R., & Saini, A. K. (2018). Deep Learning Techniques for Phishing Detection and Classification. In Proceedings of the 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA) (pp. 1-6).
6. Sinha, R., & Mohan, V. (2018). Efficient Email Spam Detection Using Deep Learning Techniques. In Proceedings of the 2018 International Conference on Communication and Signal Processing (ICCSP) (pp. 1516-1520).
7. Ayyadevara, V. S. S., & Kumar, A. A. (2017). Efficient Email Spam Classification Using Deep Learning Techniques. In Proceedings of the 2017 International Conference on Computer Communication and Informatics (ICCCI) (pp. 1-6).
8. Marinho, T., & Santos, R. (2017). Detecting Phishing Websites Using Deep Learning. In Proceedings of the 2017 IEEE/ACM 25th International Conference on Program Comprehension (pp. 313-314).