

DETECTION OF PHISHING URLs USING MACHINE LEARNING TECHNIQUES

A PROJECT REPORT

Submitted by

RAJARSHI BOSE (20BCE2574), SRIJAN SINGH SOMVANSHI
(20BDS0381), ROHAN BEJOY (20BCE2586)

Course Code: CSE3501

Course Title: Information Security Analysis and Audit

Under the guidance of
Dr. Kakelli Anil Kumar
Associate Professor
SCOPE, VIT, Vellore.



VIT[®]

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

November-2022

INDEX

	Page No.
1. Introduction	03-04
1.1. The Technique of Phishing	
1.2. Detection of Phishing URLs	
2. Literature Survey (2.1. – 2.16. Consisting of 16 Articles)	04-17
3. Overview of the Work	17-25
3.1. Problem description	
3.2. Working model	
3.3. Design description	
3.4. Algorithms	
4. Dataset Description	25-31
4.1. Data Cleaning	
4.2. Feature Selection	
4.3. Data Transformation	
4.4 Parameter	
5. Implementation	32-52
5.1. Description of Modules	
5.2. Machine Learning Algorithms/Testing	
5.3. Implementation Tools	
5.4. Test Cases	
5.5. Execution of the project	
5.6. Results analysis	
6. Conclusion and Future Scope	52
7. References	53-55

ABSTRACT

Phishing is a common attack on credulous people by making them to disclose their unique information using counterfeit websites. The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Cyber security persons are now looking for trustworthy and steady detection techniques for phishing URLs detection. This paper deals with machine learning technology for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Naive Bayes Classifier, Decision Tree, Random Forest, Gradient Boosting Classifier, CatBoost Classifier, XGBoost Classifier and Multi-layer Perceptron algorithms are used to detect phishing URLs. The objective of the paper is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, f1 score, precision and recall rate of machine learning algorithms.

1. Introduction

Phishing is a fraudulent technique that uses social and technological tricks to steal customer identification and financial credentials. Social media systems use spoofed e-mails from legitimate companies and agencies to enable users to use fake websites to divulge financial details like usernames and passwords. Hackers install malicious software on computers to steal credentials, often using systems to intercept username and passwords of consumers' online accounts. Phishers use multiple methods, including email, Uniform Resource Locators (URL), instant messages, forum postings, telephone calls, and text messages to steal user information. The structure of phishing content is similar to the original content and trick users to access the content in order to obtain their sensitive data. The primary objective of phishing is to gain certain personal information for financial gain or use of identity theft. Phishing attacks are causing severe economic damage around the world. Moreover, most phishing attacks target financial/payment institutions and webmail, according to the Anti-Phishing Working Group (APWG) latest Phishing pattern studies.

1.1. The Technique of Phishing

The criminals, who want to obtain sensitive data, first create unauthorized replicas of a real website and e-mail, usually from a financial institution or another company that deals with financial information. The e-mail will be created using logos and slogans of a legitimate company. The nature and format of Hypertext Mark-up Language makes it very easy to

copy images or even an entire website. While this ease of website creation is one of the reasons that the Internet has grown so rapidly as a communication medium, it also permits the abuse of trademarks, trade names, and other corporate identifiers upon which consumers have come to rely as mechanisms for authentication. Phisher then send the "spoofed" e-mails to as many people as possible in an attempt to lure them in to the scheme. When these e-mails are opened or when a link in the mail is clicked, the consumers are redirected to a spoofed website, appearing to be from the legitimate entity.

1.2. Detection of Phishing URLs

The general method to detect phishing websites by updating blacklisted URLs, Internet Protocol (IP) to the antivirus database which is also known as "blacklist" method. To evade blacklists attackers use creative techniques to fool users by modifying the URL to appear legitimate via obfuscation and many other simple techniques including: fast-flux, in which proxies are automatically generated to host the web-page; algorithmic generation of new URLs; etc. Major drawback of this method is that, it cannot detect zero-hour phishing attack. Heuristic based detection which includes characteristics that are found to exist in phishing attacks in reality and can detect zero-hour phishing attack, but the characteristics are not guaranteed to always exist in such attacks and false positive rate in detection is very high.

To overcome the drawbacks of blacklist and heuristics-based method, many security researchers now focused on machine learning techniques. Machine learning technology consists of many algorithms which requires past data to make a decision or prediction on future data. Using this technique, algorithm will analyze various blacklisted and legitimate URLs and their features to accurately detect the phishing websites including zero- hour phishing websites.

2. Literature Survey

2.1. Chawla, A. (2022, March). Phishing website analysis and detection using Machine Learning. International Journal of Intelligent Systems and Applications in Engineering (IJISAE), 10(1), 10–16. DOI: <https://doi.org/10.18201/ijisae.2022.262>

Abstract: With more than 820 million users of the internet by the year 2022, there is a need for security systems to protect the public from phishing scams as it not only

affects the wealth of public but also affects the mental health of public, making people afraid to surf or use the internet services, which motivates me to work on this problem to develop an efficient solution. Cybersecurity has become an essential component of this new digital age. The purpose of this study is to develop a model that can identify websites that engage in phishing by analysing certain characteristics that are shared by fraudulent websites. On the dataset, several different models were trained, including the Random Forest Classifier, the Decision Tree Classifier, Logistic Regression, K Nearest Neighbours, Artificial Neural Networks, and the Max Vote Classifier of Random Forest, Artificial Neural Networks, and K Nearest Neighbours. Max Vote Classifier of Random Forest (max depth 16), Decision Tree (max depth 18), and Artificial Neural Network all achieved an accuracy of 97.73%, with Max Vote Classifier of Decision Tree achieving the highest accuracy. Implementing a web application that allows users to enter a website link and, using that link to obtain values for a variety of factors on which the model was trained, the application will be able to determine whether or not a website is a phishing website. This research has the potential to be put into practise in the real world.

URL: <https://ijisae.org/index.php/IJISAE/article/view/1333>

- 2.2. Kumar, j., Santhanavijayan, A., Janet, B., Rajendran, B. & Bindhumadhava, B. S. (2020). Phishing Website Classification and Detection Using Machine Learning. International Conference on Computer Communication and Informatics (ICCCI), pp. 1-6, DOI: 10.1109/ICCCI48352.2020.9104161**

Abstract: Recently, the phishing website has emerged as a significant threat to the integrity of the network. The phishing websites contain a variety of malicious software, including spam, malware, ransomware, and drive-by abuses. A phishing website will frequently imitate another well-known website in order to lure unsuspecting customers into falling for the con and giving up their personal information. The person who falls victim to the tricks suffers a loss of notoriety, financial hardship, and the disclosure of private information. As a result, it is of the utmost importance to locate a solution that could effectively mitigate such security risks in a timely manner. In most cases, the identification of phishing websites is accomplished through the use of boycotts. There are many well-known websites, such as PhisTank, that provide a list of other websites

that should be avoided. The boycotting method requires two points of view because it is unlikely to be comprehensive and does not identify newly created phishing sites. Recent years have seen the implementation of AI strategies for the purpose of classifying phishing sites and finding new ones. In this paper, we investigated various AI approaches for the task of grouping phishing URLs, and we achieved the highest accuracy of 98% for the Naive Bayes Classifier with a precision = 1, review = .95, and F1-Score = .97. This result was achieved with a precision = 1, review = .95, and F1-Score = .97.

URL: <https://ieeexplore.ieee.org/document/9104161>

- 2.3. Mahajan, R. & Siddavatam, I. (2018, October). Phishing Website Detection using Machine Learning Algorithms. International Journal of Computer Applications (0975 – 8887), Volume 181 – No. 23. DOI: 10.5120/ijca2018918026**

Abstract: An attack using phishing is the simplest way to obtain sensitive information from users who are not involved. Phishers seek out victims with the intention of stealing sensitive information such as usernames, passwords, and financial account information. People working in the field of cyber security are currently looking for reliable and consistent detection methods that can identify phishing websites. The purpose of this paper is to discuss the application of machine learning technology to the problem of detecting phishing URLs by extracting and comparing the various characteristics of legitimate and fraudulent URLs. Phishing websites can be identified through the use of algorithms such as decision trees, random forests, and support vector machines. The purpose of this paper is to identify phishing URLs and to determine which machine learning algorithm is the most effective by contrasting the accuracy rate, the number of false positives, and the number of false negatives produced by each algorithm.

URL:

https://www.researchgate.net/publication/328541785_Phishing_Website_Detection_using_Machine_Learning_Algorithms

- 2.4. **Dutta, A. K. (2021, October). Detecting phishing websites using machine learning technique. PLoS ONE 16(10): e0258361. DOI: <https://doi.org/10.1371/journal.pone.0258361>**

Abstract: In recent years, advancements in Internet and cloud technologies have led to a significant increase in electronic trading, in which consumers make purchases and transactions online. This has led to an increase in the number of people engaging in electronic trading. This growth results in unauthorised access to the sensitive information of users and causes damage to an enterprise's resources. Phishing is a common form of online attack that involves deceiving users into visiting malicious websites in order to steal their personal information. The majority of fraudulent websites have interfaces and uniform resource locators (URLs) that are virtually indistinguishable from those of legitimate websites. Many different methods, such as blacklists, heuristics, and others, have been proposed as potential solutions to the problem of phishing websites. Despite this, there has been an exponential increase in the number of victims, which can be attributed to ineffective security technologies. Phishing attacks are more likely to be successful on the Internet because of its anonymous and uncontrollable infrastructure. The performance of the phishing detection system is shown to be limited by the research works that are currently available. Users are looking for an intelligent method that can protect them from cyberattacks, and this demand is growing. An approach to URL detection that was proposed by the author of this study and was based on machine learning techniques. In order to identify potentially fraudulent URLs, a technique based on recurrent neural networks is utilised. The researcher assessed the effectiveness of the proposed method using 7900 malicious sites and 5800 legitimate sites, respectively. The results of the experiments indicate that the performance of the proposed method in identifying malicious URLs is superior to the recent approaches that have been used.

URL: <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0258361>

- 2.5. **Sampat, H., Saharkar, M., Pandey, A. & Lopes, H. (2018, March). Detection of Phishing Website Using Machine Learning. International Research Journal of Engineering and Technology (IRJET), Volume: 05 Issue: 03. DOI: <https://irjet.net/archives/V5/i3/IRJET-V5I3580.pdf>**

Abstract: The detection of phishing websites is accomplished through the utilisation of classification or association Data Mining algorithms, which results in a model that is both intelligent and efficient. These algorithms were used to identify and characterise all of the rules and factors that need to be classified in order to classify the phishing website and the relationship that correlates them with each other so that we can detect them based on their performance, accuracy, number of rules generated, and speed. The proposed system utilises both the Classification and Association optimization algorithms, making it both more effective and quicker than the currently in place system. The error rate of the existing system is reduced by thirty percent when these two algorithms and the WHOIS protocol are used together; consequently, the proposed system creates an effective way to identify phishing websites by making use of this method. In spite of the fact that there is no known method that can identify each and every website that engages in phishing, putting these strategies into practise will result in the development of the most effective method for identifying phishing websites.

URL:

https://www.academia.edu/36834032/Detection_of_Phishing_Website_Using_Machine_Learning

- 2.6. Deshpande, A., Pedamkar, O., Chaudhary, N. & Borde, S. (2021, May). Detection of Phishing Websites using Machine Learning. International Journal of Engineering Research & Technology (IJERT), Vol. 10 Issue 05. DOI: 10.17577/IJERTV10IS050235**

Abstract: A common form of social engineering, phishing targets trusting individuals and coerces them into divulging their personal information by means of bogus websites. Phishing website URLs have the purpose of stealing personal information such as user names, passwords, and financial transactions made online in order to commit identity theft. Phishers create websites that are visually and semantically similar to the websites they are targeting, and then direct their victims to those fake websites. Phishing techniques started to advance at a rapid pace as technology continued to advance, and this needs to be prevented by using anti-phishing

mechanisms to detect phishing. The fight against phishing can be aided significantly by using machine learning, which is a powerful tool. This paper provides an overview of the features that are used for detection as well as the detection techniques that use machine learning. Phishing is a popular method of attack among cybercriminals because it is simpler to deceive a target into clicking on a malicious link that appears to be legitimate than it is to penetrate a computer's security systems. The body of the message contains malicious links that are designed to give the impression that clicking on them will take the recipient to the spoofed organisation by using the logos and other legitimate content associated with the spoofed organisation. In this article, we will explain the characteristics of phishing domains, also known as fraudulent domains, as well as the characteristics that differentiate them from legitimate domains, why it is important to detect these domains, and how machine learning and natural language processing techniques can be used to detect them.

URL: <https://www.ijert.org/detection-of-phishing-websites-using-machine-learning>

- 2.7. Pratik, N. N., Vaneeta, M., Prajwal, D., Pradeep, K. S. & Kakade, S. K. (2020, June). Detection of Phishing Websites Using Machine Learning Techniques. International Journal of Emerging Technologies and Innovative Research (JETIR), Vol.7, Issue 6, page no.117-123. DOI: <http://www.jetir.org/papers/JETIR2006018.pdf>**

Abstract: Forging a website with the intention of tracking and stealing the sensitive information of online users is an activity known as phishing. It is a form of identity theft in which criminals construct replicas of target websites and lure victims into divulging sensitive information such as passwords, PIN numbers, and the like. The amount of data that is downloaded from and uploaded to the web on a regular basis is staggering. Criminals will have more opportunities to hack into important personal information as a result of this. We have developed a method for the detection of phishing websites that is based on machine learning classifiers and incorporates a wrapper features selection method in order to address the challenges that have been presented here. Artificial Neural Network, Random Forest, and Support Vector Machine are some of the classification algorithms that are utilised. Dynamic features

are extracted from the URL that was entered, and a trained model is used to determine whether or not the URL in question is a phishing site.

URL: <https://www.jetir.org/view?paper=JETIR2006018>

- 2.8. Patil, V., Thakkar, P., Shah, C., Bhat, T. & Godse, S.P. (2018). Detection and Prevention of Phishing Websites Using Machine Learning Approach. Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), pp. 1-5. DOI: 10.1109/ICCUBEA.2018.8697412**

Abstract: Internet users lose a lot of money every year due to the scam known as phishing. It makes reference to exploiting a shortcoming on the customer's side, which renders them defenceless against such attacks. Due to the magnitude of the phishing problem and the absence of a single solution that can effectively address all of its vulnerabilities, various strategies have been put into practise. In this article, we discuss three different approaches that can be utilised to identify phishing websites. The first method involves dissecting various aspects of the URL, the second method involves verifying the legitimacy of the website by determining where the website is hosted and who is in charge of it, and the third method employs an investigation that is based on the website's outward appearance to validate the website. In order to evaluate these various highlights of URLs and websites, we make use of the procedures and calculations that are associated with machine learning. An overview of these methodological approaches is provided in the current work here.

URL:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8697412&isnumber=8697217>

- 2.9. Garcés, I. O., Cazares, M. F. & Andrade, R. O. (2019). Detection of Phishing Attacks with Machine Learning Techniques in Cognitive Security Architecture. International Conference on Computational Science and Computational Intelligence (CSCI), pp. 366-370. DOI: 10.1109/CSCI49370.2019.00071**

Abstract: The number of phishing attacks has increased in Latin America to a level that is beyond the capabilities of the network security examiners who work there. The application for psychological protection suggests making use of big data, artificial intelligence, and information analysis in order to speed up response times in potential attack zones. This paper presents an investigation about the investigation of atypical behaviour related with phishing web assaults and how AI procedures can be an alternative to confront the issue. Specifically, the paper focuses on how AI procedures can be an alternative to confront the issue. This investigation makes use of a tainted informational collection and python tools for the creation of artificial intelligence for identifying phishing attacks. This is done by analysing URLs to determine whether they are fortunate or unfortunate URLs based on the specific attributes of the URLs. The purpose of this investigation is to provide real-time data that can be used to make proactive decisions that reduce the impact of an attack.

URL: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9070902&isnumber=9070327>

- 2.10. Niakanlahiji, A., Chu, B. -T. & Al-Shaer, E. (2018). PhishMon: A Machine Learning Framework for Detecting Phishing Webpages. IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 220-225. DOI: 10.1109/ISI.2018.8587410**

Abstract: Phishing attacks continue to be common and highly effective in their goal of getting unsuspecting users to reveal sensitive information such as account credentials and social security numbers. This is despite the fact that numerous research efforts have been conducted on the subject. In this article, we present PhishMon, a new machine learning framework that is loaded with features and is designed to identify phishing websites. It is based on a set of fifteen innovative features that can be efficiently computed from a webpage and does not rely on any third-party services, such as WHOIS servers or search engines. These features capture a variety of distinguishing characteristics of legitimate web applications as well as the web infrastructures upon which they are built. Phishers incur additional costs when they attempt to emulate these features because doing so requires them to expend a significant amount of additional time and effort on their underlying infrastructures and

web applications. This is in addition to the work that must be done to replicate the appearance of target websites. We demonstrate that PhishMon is capable of distinguishing unseen phishing websites from legitimate websites by conducting extensive testing on a dataset that contains 4,800 unique phishing websites and 17,500 unique benign websites. This demonstrates that PhishMon has a very high degree of accuracy. In the tests that we ran, PhishMon had an accuracy of 95.4%, with only a 1.3% rate of false positives, when applied to a dataset that contained unique phishing instances.

URL:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8587410&isnumber=8587309>

- 2.11. Alswailem, A., Alabdullah, B., Alrumayh, N. & Alsedrani, A. (2019). Detecting Phishing Websites Using Machine Learning. International Conference on Computer Applications & Information Security (ICCAIS), pp. 1-6. DOI: 10.1109/CAIS.2019.8769571**

Abstract: One of the web security issues that focuses on human weaknesses rather than programming weaknesses is the phishing site. Phishing sites are one example. It is possible to portray it as a strategy for luring online customers in order to acquire their sensitive information, such as usernames and passwords. This is one interpretation. In this article, we present a savvy framework that can be used to identify phishing websites. The system acts as an additional feature that can be added to a web application in the form of an expansion, and it notifies the user immediately whenever the system detects a phishing site. A method of artificial intelligence, specifically directed learning, is required for the framework. Because of its excellent presentation in characterization, the Random Forest procedure is the one that we have decided to use. Examining the highlights of the phishing site and selecting the most effective combination of those highlights to create the classifier is our primary focus as we work to improve the classification system. Following that, we completed our paper with a precision of 98.8% and a variety of 26 highlights.

URL:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8769571&isnumber=8769442>

- 2.12. Abdelhamid, N., Thabtah, F. & Abdel-jaber, H. (2017). Phishing detection: A recent intelligent machine learning comparison based on models' content and features. IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 72-77. DOI: 10.1109/ISI.2017.8004877**

Abstract: In the most recent decade, numerous fake websites have been developed on the World Wide Web to imitate restricted sites, with the purpose of stealing financial resources from customers and organisations. Phishing is the name given to this type of attack carried out over the internet, and it is estimated that it has cost the online network and its various partners several million dollars. In light of this, effective countermeasures that are capable of discriminating phishing in an exact manner are required. AI (ML) is a well-known tool for information investigation, and as of late, it has indicated promising outcomes in the fight against phishing when looked at differently in comparison to exemplary enemies of phishing approaches, such as mindfulness workshops, representation, and legal arrangements. This article investigates the appropriateness of using ML procedures to differentiate between phishing attacks and depicts the benefits and drawbacks of using them. To be more specific, many different kinds of ML methods have been investigated in order to discover the appropriate choices that can act as safeguards against phishing tools. Moreover, and this is of the utmost importance, we perform a preliminary investigation of a huge number of ML strategies on genuine phishing datasets and in relation to a variety of measurements. The purpose of the investigation is to discover the strengths and weaknesses of ML predictive models and to demonstrate how well they perform in relation to phishing assaults. The results of the research indicate that Covering approach models are superior to phishing arrangements, particularly for new customers. This is due to the fact that Covering approach models have information bases that are simple yet powerful, and they also have a high phishing recognition rate.

URL:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8004877&isnumber=8004858>

- 2.13. **Das Gupta, S., Shahriar, K.T., Alqahtani, H. et al (2022, March). Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques. Ann. Data. Sci. DOI: <https://doi.org/10.1007/s40745-022-00379-8>**

Abstract: This paper focuses primarily on presenting an approach to the detection of real-time phishing websites that is based on machine learning. This approach takes into account URL and hyperlink-based hybrid features to achieve high accuracy without relying on any third-party systems. The goal of phishing is to steal sensitive information such as usernames, passwords, social security numbers, credit card information, and so on from internet users. Phishers typically try to trick internet users by disguising a webpage as an official genuine webpage in order to steal this information. Anti-phishing solutions such as blacklist or whitelist, heuristic, and visual similarity-based methods are unable to detect zero-hour phishing attacks or freshly launched websites. In addition, earlier methods are difficult to implement and are not appropriate for use in real-time environments because they rely on external sources, such as a search engine, for their data. Therefore, one of the greatest challenges in the field of cybersecurity is the detection of recently developed phishing websites in an environment that operates in real time. This paper proposes a hybrid feature-based anti-phishing strategy, which extracts features from the URL and hyperlink information on the client-side only. The goal of this strategy is to solve the problems described above. In addition to this, we create a brand-new dataset with the intention of using well-known machine learning classification strategies as a basis for our experiments. According to the findings of our experiments, the recently proposed method for detecting phishing attacks is superior to more conventional methods in terms of effectiveness, boasting a detection accuracy of 99.17% when using the XG Boost technique.

URL: <https://link.springer.com/article/10.1007/s40745-022-00379-8#citeas>

- 2.14. Chatterjee, M. & Namin, A. -S. (2019). Detecting Phishing Websites through Deep Reinforcement Learning. IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), pp. 227-232. DOI: 10.1109/COMPSAC.2019.10211

Abstract: Phishing is a type of cybercrime that requires the least amount of skill and has the goal of convincing individuals to part with sensitive data such as exclusively conspicuous information, banking and Visa nuances, orev credentials and passwords. Phishing is the type of cybercrime that requires the least amount of skill and has the goal of convincing individuals to part with sensitive data. Typically, this type of fundamental yet effective digital attack is delivered via messages, phone calls, or texts. However, it can also be delivered via email. The stolen certification or private information is then used to gain access to basic records of the people who are the subject of the investigation, which can result in widespread blackmailing and financial hardship. As a direct consequence of this, one of the steps in the process of phishing involves the sending of retaliatory messages to individuals who have been targeted. The majority of the time, a phisher will construct a fraudulent website with the intention of duping users into divulging their login credentials and other sensitive information when they are not expecting it. It is essential, prior to inflicting any kind of pain or suffering on innocent people, to locate the malicious websites that have the potential to bring about said pain and suffering. In this paper, a novel approach that is based on extensive research is presented for the purpose of figuring out how to display and differentiate malicious URLs. The growing phenomenon of phishing sites provided the impetus for the idea that ultimately led to the creation of this paper. The model that has been proposed is appropriate for adjusting to the dynamic behaviour of phishing sites and, as a consequence of this, becoming familiar with the prominent characteristics that are associated with the discovery of phishing sites.

URL:

<https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8754075&isnumber=8753906>

- 2.15. Selvakumari, M & Sowjanya, M & Das, Sneha & Padmavathi, S. (2021). Phishing website detection using machine learning and deep learning techniques. Journal of Physics: Conference Series. 1916. 012169. DOI: 10.1088/1742-6596/1916/1/012169.**

Abstract: As the number of people who use the internet continues to rapidly increase, phishing attacks have become increasingly damaging. The phishing attack is currently a significant risk to both people's day-to-day lives and the environment of the internet. An attacker will impersonate a trusted entity in order to carry out these types of attacks. The attacker's goal is to steal sensitive information or the digital identity of the user, such as account credentials, credit card numbers, and other details about the user. A website is said to be phishing if it has a name and an appearance that are very similar to those of an official website. This type of website is also known as a spoofed website and is designed to trick users into providing their personal information so that the perpetrators can steal it. This paper will discuss the machine learning and deep learning algorithms and apply all of these algorithms on our dataset. Then, the best algorithm having the best precision and accuracy will be selected for the phishing website detection. So, to identify the websites that are fraudulent, this paper will discuss the machine learning and deep learning algorithms. This work has the potential to provide more efficient defences against phishing attacks in the future.

URL:

https://www.researchgate.net/publication/351929313_Phishing_website_detection_using_machine_learning_and_deep_learning_techniques

- 2.16. Mamun, M. S. I., Rathore, M. A., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016, September). Detecting malicious urls using lexical analysis. In International Conference on Network and System Security (pp. 467-482). Springer, Cham.**

Abstract: Since quite some time ago, the Internet has developed into an important venue for committing crimes online. In this particular sector, URLs serve as the primary mode of transportation. In order to combat these problems, the security network has redirected its efforts to the development of strategies for the widespread

boycotting of malicious URLs. Although this strategy is effective in protecting customers from known dangerous environments, it only addresses one of the factors that contribute to the problem. The new malicious URLs that have sprung up all over the internet in large numbers on a consistent basis have a head start in this competition. Aside from that, Alexa believed sites might pass on undermined fake URLs known as disfigurement URL. In this work, we investigate a lightweight method for dealing with the identification and order of the malicious URLs according to the type of attack that they carry out. We demonstrate that lexical investigation can be successful and fruitful when it comes to the proactive discovery of URLs by using these URLs. We present an arrangement of adequate details that are required for exact order and evaluate the accuracy of the methodology by applying it to a collection of more than 110,000 URLs. In addition, we investigate the effect that obscurity strategies have on harmful URLs in order to identify the type of jumbling strategy that is most effective against particular categories of harmful URL.

URL: <https://www.springerprofessional.de/en/detecting-malicious-urls-using-lexical-analysis/10722098>

3. Overview of the Work

3.1. Problem description

Phishing is one of the techniques which are used by the intruders to get access to the user credentials or to gain access to the sensitive data. This type of accessing the is done by creating the replica of the websites which looks same as the original websites which we use on our daily basis but when a user clicks on the link, he will see the website and think its original and try to provide his credentials. URLs sometimes known as “Web links” are the primary means by which users locate information in the Internet. To overcome this problem some of the machine learning techniques are used for detection of phishing URLs by extracting and analyzing various features of legitimate and phishing URLs. Our aim is to detect phishing URLs as well as narrow down to best machine learning algorithm by comparing accuracy rate, f1 score, precision and recall rate of machine learning algorithms.

3.2. Working model

The process begins with acquiring of the raw dataset from the Kaggle repository. This gathered raw dataset is preprocessed to improve its reliability and usefulness for the machine learning algorithms to learn from this dataset. Data cleansing is carried out to remove junk or missing values or invalid values that could cause difficulty to digest by the Machine Learning Algorithms. Better the data, easier it is for the Machine Learning algorithms to produce better results.

In the next stage Machine Learning models are built by making use of versatile algorithms. Algorithms used are Logistic Regression, K-Nearest Neighbors, Support Vector Machine, Naive Bayes Classifier, Decision Tree, Random Forest, Gradient Boosting Classifier, CatBoost Classifier, XGBoost Classifier and Multi-layer Perceptron. Each algorithm has its advantages and is harnessed in producing the final outcome. Each model would produce its output i.e., its prediction about whether the passed data has features that could be either a phished URL or a legitimate one. Next, we are comparing the results produced and the most effective algorithm is chosen to be the best model. Further to enhance its performance and reliability Dynamic features extraction is performed to obtain the dataset of newly entered URL by the user and the algorithm is able to produce accurate results for newly entered data. Finally, as an end result the software that makes use of the above process must be able to predict whether the URL of the website that the user wishes to access is Phishing or Legitimate. Figure 1 shows the architecture of the proposed model.

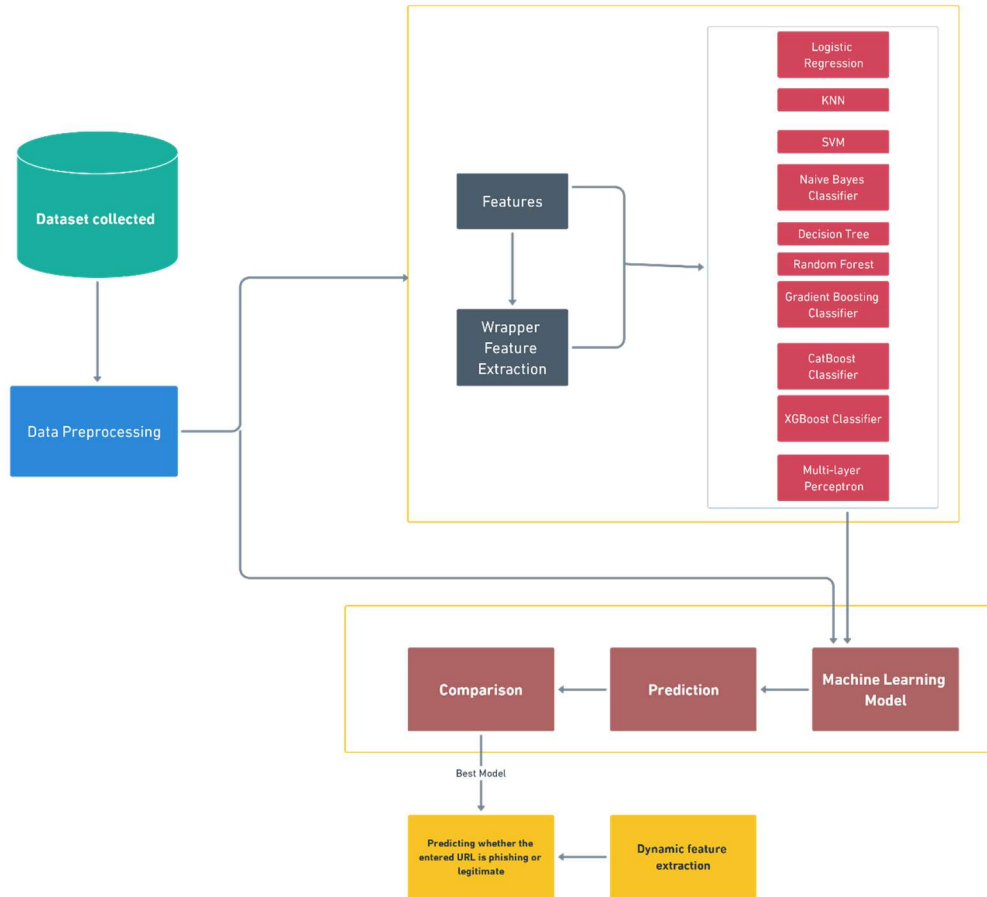


Figure 1: Architecture of the proposed system

We have developed our project using a website as a platform for all the users. This is an interactive and responsive website that will be used to detect whether a website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS and JavaScript. The basic structure of the website is made with the help of HTML. CSS is used to add effects to the website and make it more attractive and user-friendly. It must be noted that the website is created for all users, hence it must be easy to operate with and no user should face any difficulty while making its use. Every naïve person must be able to use this website and avail maximum benefits from it.

3.3. Design description

The work consists of Address Bar based, Abnormal Based, HTML and JavaScript Based and Domain Based Features extraction of collected URLs and analysis. The first

step is the collection of phishing and benign URLs and then data preprocessing is done. The Address Bar based, Abnormal Based, HTML and JavaScript Based and Domain Based feature extractions are applied to form a database of feature values. The database is knowledge mined using different machine learning models and then the different machine learning models are compared based on the accuracy of the model and the best model is selected and that model is used for predicting whether the URL entered is a Phishing URL or a Legitimate URL. Figure 2 shows the design flow graph.

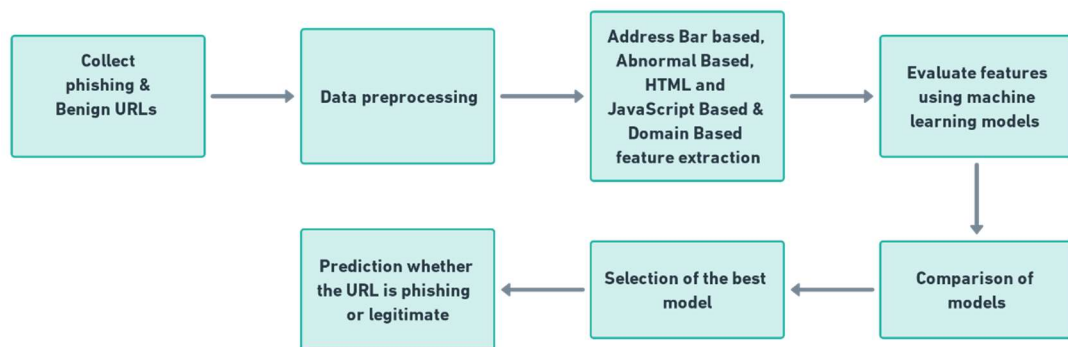


Figure 2: Design flow graph

3.4. Algorithms

3.4.1. Data collection

Input: Data Repositories

Output: Raw Data

1. procedure DATA COLLECTION
2. $W \leftarrow \text{ExtractData}(\text{Repositories})$
3. $W1 \leftarrow \text{FilterInvalidURL}(W)$
4. $N \leftarrow \text{Count}(W1)$
5. **return** $W1, N$
6. **end** procedure

3.4.2. Data Pre-process

Input: Raw Data

Output: Features

```
1. procedure DATA PRE-PROCESS
2.   while  $url \leftarrow URL$  do
3.      $D \leftarrow \text{RemoveProtocols}(\text{RawData})$ 
4.      $D1 \leftarrow \text{RemoveIrrelevant}(D)$ 
5.      $D2 \leftarrow \text{RemoveDomainName}(D1)$ 
6.   end while
7.   return  $D2$ 
8. end procedure
```

3.4.3. Data transformation

Input: Features($D2$)

Output: Vectors

```
1. procedure DATA TRANSFORMATION( $D2$ )
2.   while  $d \leftarrow D2$  do
3.      $Num \leftarrow \text{GenerateVectors}(d)$ 
4.   end while
5.   return  $Num$ 
6. end procedure
```

3.4.4. Training Phase

Input: Vector (Num), URL

Output: URL-Type

```
1. procedure TRAINING PHASE(Num)
2.   while  $num \leftarrow Num$  do
3.     if  $num = \text{Feature(URL)}$  then
4.        $op = \text{Phishing URL}$ 
5.     else
6.        $op = \text{Legitimate URL}$ 
7.       if  $op = \text{LSTMLib(feature)}$  then
8.          $op = \text{Phishing URL}$ 
9.       else
10.         $op = \text{Legitimate URL}$ 
11.      end if
12.    end if
13.  end while
14.  return  $op$ 
15. end procedure
```

3.4.5. Testing Phase

Input: URL

Output: Type of URL

```
1. procedure TESTING PHASE(URL)
2.   while  $url \leftarrow URL$  do
3.     if  $element \leftarrow LST\ M\ Memory = Feature(URL)$  then
4.        $op = \text{Phishing URL}$ 
5.     else
6.        $op = \text{Legitimate URL}$ 
7.        $feedback = phishtank(op)$ 
8.       if  $element \leftarrow LST\ M\ Memory = f \leftarrow feedback$  then
9.          $op = \text{Phishing URL}$ 
10.      else
11.         $op = \text{Legitimate URL}$ 
12.      end if
13.    end if
14.  end while
15.  return  $op$ 
16. end procedure
```

3.4.6. Extract the URL based features

Input: URL of suspicious website U

Output: Character sequences vector $F_U = \langle c_1, c_2, c_3, \dots, c_{200} \rangle \in F_1$

Start

1. Initialize Tokenizer (char_level=True, oov_token='UNK'),
Alphabet="abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789,;.!?:'\"/\\|_@#\$\$%^&*~`+-=<>{}"char_dict = { }
2. **for** $i, char$ in enumerate(Alphabet) **do**
3. char_dict[char] = i + 1
4. Tokenizer.word_index = char_dict
5. **end for**
Add 'UNK' to the vocabulary
6. Tokenizer.word_index[oov_token] = max_value_of_char_dict + 1
7. $U_character_sequences = \text{Tokenizer.texts_to_sequences}(U)$
8. **If** len($U_character_sequences$) < 200 **then**
the remaining part will be filled as 0
9. $F_U = \langle c_1, c_2, c_3, \dots, 0, 0, 0, c_{200} \rangle$
10. **else**
the part longer will be truncated
11. $F_U = \langle c_1, c_2, c_3, \dots, c_{200} \rangle$
12. **end if**
13. Return F_U

End

3.4.7. Extract the textual content feature of the webpage

Input: A HTML document doc

Output: TF-IDF vector N-gram chars $F_T = \langle t_1, t_2, t_3, \dots, t_D \rangle \in F_2$

Start

1. $P_T = \text{getPlaintext}(doc)$
2. $N_T = \text{getTagAttributesValues}(doc)$
(DIV, IMG, Body, Footer, a, Link, Article, Label,
H1...H5, Template...etc.)
3. $T_1 = P_T \cup N_T$
4. Text cleaning and preprocessing
 - $T_2 = \emptyset, T_3 = \emptyset$
 - $T_2 = \text{Text_cleaner}(T_1)$
(Remove punctuations symbols, numbers, spaces,
newline, character that are not English)
 - **for** $token$ in T_2 **do**
 - **if** $token$ not in STOPWORDS and len($token$) > 3 **then**
 - $T_3 = T_3 \cup \text{lemmatize_stemming}(token)$
 - **end if**
 - **end for**
5. $F_T = \text{TF-IDF_Ngram_chars_Transform}(T_3)$
6. Return F_T

End

3.4.8. Phishing Website Detection Model

Input: URL of suspicious website

Output: $Prediction \in \{1, -1\}$, $-1 \rightarrow$ phishing, $1 \rightarrow$ legitimate

- 1: Procedure PhishDetection(*input URL*);
- 2: Rule generation for URL features
- 3: Extract URL based features ($UF1 - UF15$)
- 4: Extract HTML source code & create DOM Tree
- 5: Rule generation from DOM tree for hyperlink features
- 6: Extract hyperlink based features($HF1 - HF10$) from DOM Tree
- 7: Generate hybrid feature set by combining URL based and hyperlink based features
- 8: Remove unuseful feature $UF1$
- 9: Apply hybrid feature set on well performed machine learning classifier
- 10: **if** classifier predicts URL as phishing **then**
- 11: Prediction $\leftarrow -1$
- 12: **else**
- 13: Prediction $\leftarrow 1$
- 14: **end if**
- 15: **return** Prediction

4. Dataset description

Phishing website dataset provided by Kaggle repository is used for this project work. Dataset consists of a collection of website URLs for 11000+ websites with 30 parameters about each website, each parameter has value either -1, 0, 1 and a class label which signifies whether the website tested is a phishing website or not (1 or -1). Dataset is sliced into 8:2 ratio where 80% websites were used for training the model and 20% were used for testing the model which is trained.

4.1. Data Cleaning

The dataset doesn't contain any missing value or outlier value hence there was no modifications were made on the dataset.

4.2. Feature Selection

Wrapper based feature selection was used to try out all the possible combinations and highest accuracy was achieved when we used all the 30 parameters.

4.3. Data Transformation

Data transformation was not used as data was already defined in 3 integers set of {-1, 0, 1}.

4.4. Parameter

The dataset has over 30 parameters.

4.4.1. Having IP Address

This parameter signifies whether the website registered domain name or not as non registered domain websites will be shown in form IP address with specified port number in the search bar. This parameter can have only 2 possible values either 1 or -1 where -1 for having IP address.

4.4.2. URL Length

This parameter signifies about length of the URL if the characters length is less than 54 then website is considered legitimate website and value of attribute is 1 if the size is greater than 54 and less than equal to 75 is considered suspicious and value of attribute is 0 and more than 75 is considered phishing website and value of attribute is -1.

4.4.3. URL Shortening

This parameter signifies whether the website link using is shortened which can redirect to Phishing websites. This parameter has 2 values -1 and 1 where -1 for using shortening service.

4.4.4. Having '@' symbol

This parameter signifies whether the website link has "@" symbol in the character set of the link as browsers tend to ignore address before "@" symbol and which could lead to a Phishing site. This parameter can have values -1 and 1 where -1 means website link has "@" symbol.

4.4.5. Using ‘//’ for redirecting

This parameter signifies whether the website link has “//” in the character set of the link as browser will redirect to page mentioned after “//” so the link has to be checked for last occurrence of “//” for pages which have http or https protocol the occurrence is 6th position and 7th position respectively if it occurs after 7th position then the site is phishing site and parameter value will be given -1 else 1.

4.4.6. Addition of prefix or suffix in URL link using ‘-’

This parameter signifies whether the website link has “-” symbol which can be used to add prefix, suffix to site leading user to believe it is a genuine site, this parameter can have 2 values -1 and 1 where -1 for having “-” in URL.

4.4.7. Having Multiple subdomains

This parameter signifies whether the website is Phishing or not on basis of subdomains where removal of top-level domain and second level domain and ‘www’ subdomain and then checking number of dots left in domain. If there is less than one dot then site is non-Phishing if 1 then site is suspicious and if greater than 1 then phishing site and parameter value is given 1, 0, -1 respectively.

4.4.8. HTTPS

This parameter signifies whether the website is using HTTPS protocol or not and whether the issued certificate is 1 year old and issued by trusted authority. If HTTPS is one year old and issued by trusted authority parameter value is 1 else HTTPS issued by suspicious authority then value 0 else non HTTPS site parameter value -1.

4.4.9. Domain Registration Length

This parameter signifies whether the website domain is registered for how much more years/months as phishing sites are made for short period of time so if it is registered for less than 1 year then parameter value -1 else 1.

4.4.10. Favicon

This parameter signifies whether the website is using favicon from any other website then parameter value is given -1 and website is considered Phishing.

4.4.11. Using Non- standard port

This parameter signifies whether the site is using standard ports like for HTTPS, FTP etc. Website uses non standard ports then site is classified Phishing and parameter value is -1 as any open port not blocked by firewall can give access of target computer to the hacker.

4.4.12. HTTPS added in domain

This parameter signifies whether the site has added HTTPS in domain name to fool the target if yes then parameter value is -1 else 1.

4.4.13. Request URL

This parameter signifies whether the images, videos or any graphical content on the website is from any other website or not. In this parameter the percentage of content copied if less than equal to 22% site classified non-Phishing and parameter is given value 1 if site has more than 22% and less than 61% then site is classified suspicious and parameter is given value 0, and if more than 61% site is classified Phishing and parameter is given value -1.

4.4.14. URL of Anchor tag

This parameter signifies whether the website anchor tags point towards different domain than this website, if 31% or less anchor tags point to different domains site is classified genuine and parameter value 1, if more than 31% and less than equal to 67% site is classified suspicious and parameter value is 0 and if more than 67% website is classified Phishing and parameter value -1.

4.4.15. Links in <Meta>, <Script> and <Link> tags

This parameter signifies whether the website tags have link to the same domain of the website or different domain, if less than equal to 17% point to different link then site is classified genuine and parameter values is 1, if more than 17% and less than equal to 81% then website is classified suspicious and parameter value is 0 and more than 81% is classified Phishing with parameter value -1.

4.4.16. Server Form Handler

This parameter signifies what is value of SFH if it contains “about: blank” which means it does not define where the submitted information will be handled, there can be cases where external domains are mentioned in it. If website has same domain name in SFH then it is classified genuine with parameter value 1, if external domain is mentioned then it is classified as suspicious and parameter value is 0, and if about: blank is mentioned then site is classified Phishing with parameter value -1.

4.4.17. Submitting information to email

This parameter signifies whether there is mail () or mailto () function defined as it can lead to direct transmission of data to Phisher email, if these functions are present then parameter value is -1 else 1.

4.4.18. Abnormal URL

This parameter signifies whether the host’s name is included in the URL or not, if not then it is classified Phishing with parameter value -1 else 1.

4.4.19. Website Forwarding

This parameter signifies how many times the website has been redirected if more than once then website is classified as Phishing with parameter value -1 else 1.

4.4.20. Customized Status Bar

This parameter signifies whether the status bar show correct link or not when we hover our mouse over it, it can be checked what happens in “OnMouseOver” event if it changes then it is classified Phishing with parameter value -1 else 1.

4.4.21. Disabled Right Click

This parameter signifies whether right click is disabled or not as Phisher doesn’t wants user to inspect site and check source code, if it is disabled then parameter value is -1 else 1.

4.4.22. Using Pop-up Window

This parameter signifies whether the site is asking to fill details in pop-up window if yes then it is classified as Phishing with parameter value -1 else 1.

4.4.23. Redirection using IFRAME

This parameter signifies whether the site is using iframe to display another page without using borders which can fool the target so then the website is classified Phishing with parameter value -1 else 1.

4.4.24. Age of Domain

This parameter signifies whether the website is at least 6 months old as many Phishing sites are made for short period of time so if site is less than 6 months old then it will be classified Phishing with parameter value -1 else 1.

4.4.25. DNS Record

This parameter signifies whether DNS exist for the website if yes then genuine site else it is classified Phishing with parameter value -1 else 1.

4.4.26. Website Traffic

This parameter signifies about the popularity of the website. In Alexa Database if rank is less than 100,000 then site is considered genuine with parameter value 1, if it is more than 100,000 then website is considered suspicious with parameter value 0 and if it is not mentioned then classified Phishing with parameter value -1.

4.4.27. Page Rank

This parameter signifies how much importance of the webpage is on the internet this value is between 0 and 1 so if page rank less than 0.2 then website is Phishing and parameter value -1 else 1.

4.4.28. Google Index

This parameter signifies whether the site is indexed by Google or not. If it is not indexed, then classified as Phishing with parameter value -1 else 1.

4.4.29. Links pointing to the webpage

This parameter signifies number of links pointing to the page if it is less than 1 then considered Phishing with parameter value -1 and if more than 0 and less than 3 then website is suspicious with parameter value 0 and else it is considered genuine with parameter value 1.

4.4.30. Statistical report-based feature

This parameter signifies whether the host of website belongs to top Phishing IP or Phishing domain then website is classified with Phishing and value is -1 else 1.

Figure 3 shows the correlation of features in the dataset.

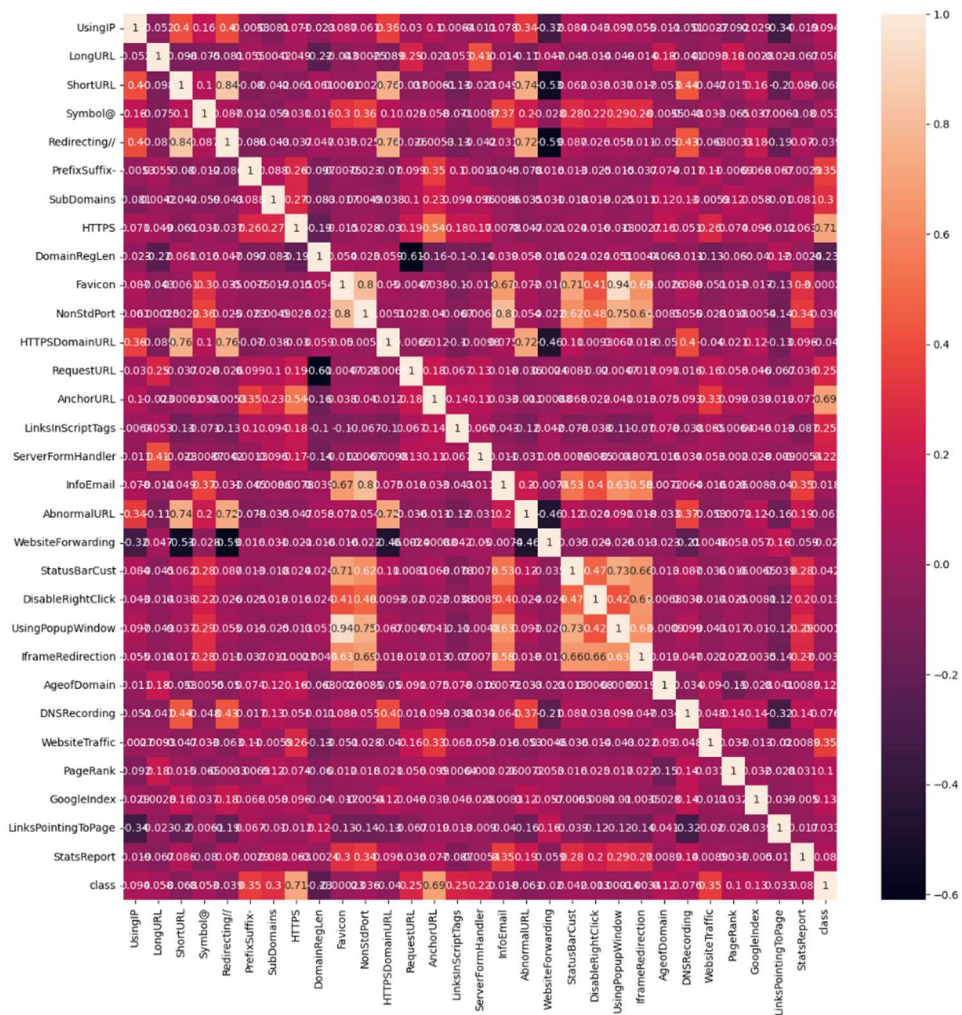


Figure 3: Correlation of features in the dataset

5. Implementation

5.1. Description of Modules

We have developed a number of modules that are capable of recognising the various components that, when present or absent, indicate that a URL may be used for phishing. This is accomplished through the use of a few different functions, each of which examines the URL. These functions will then give one of three ratings: 1 indicates that the website is safe, 0 indicates that the website is suspicious, and -1 indicates that the website in question is suspicious. The data from such modules are analysed by the machine learning model, which then assigns a score indicating whether or not the website in question is a phishing site or a website with the potential to be legitimate.

5.1.1. UsingIp(self) function

This function was developed by us in order to check whether or not the link being evaluated contains an IP address.

$$\text{Rule: IF} \begin{cases} \text{If The Domain Part has an IP Address} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.2. longUrl(self) function

This function was made to check whether the link of the website is too long.

$$\text{Rule: IF} \begin{cases} \text{URL length} < 54 \rightarrow \text{feature} = \text{Legitimate} \\ \text{else if URL length} \geq 54 \text{ and } \leq 75 \rightarrow \text{feature} = \text{Suspicious} \\ \text{otherwise} \rightarrow \text{feature} = \text{Phishing} \end{cases}$$

5.1.3. shortUrl(self) function

This function was made to check whether the link of the website has been shortened.

$$\text{Rule: IF} \begin{cases} \text{TinyURL} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.4. symbol(self) function

This function was made to check for the "@" symbol in the URL.

$$\text{Rule: IF } \begin{cases} \text{Url Having @ Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.5. redirecting(self) function

This function was made to check whether the link contains "//".

$$\text{Rule: IF } \begin{cases} \text{ThePosition of the Last Occurrence of "//" in the URL} > 7 \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.6. prefixSuffix(self) function

This function is used to check for the presence of "-" symbol in the URL.

$$\text{Rule: IF } \begin{cases} \text{Domain Name Part Includes (-) Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.7. SubDomains(self) function

This function is used to check the number of "." In the address which usually signifies the number of domains present in the URL.

$$\text{Rule: IF } \begin{cases} \text{Dots In Domain Part} = 1 \rightarrow \text{Legitimate} \\ \text{Dots In Domain Part} = 2 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

5.1.8. Hppts(self) function

This function was made to check the presence of the HTTPS protocol in the URL.

Rule:

$$\text{IF } \begin{cases} \text{Use https and Issuer Is Trusted and Age of Certificate} \geq 1 \text{ Years} \rightarrow \text{Legitimate} \\ \text{Using https and Issuer Is Not Trusted} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

5.1.9. DomainRegLen(self) function

This function was made to find out the expiry date of the domain.

$$\text{Rule: IF} \begin{cases} \text{Domains Expires on } \leq 1 \text{ years} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.10. Favicon(self) function

This function has been made to check whether the favicon has been loaded from an external domain or not.

$$\text{Rule: IF} \begin{cases} \text{Favicon Loaded From External Domain} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.11. NonStdPort(self) function

This function has been made to check whether the port # is of the preferred status or not.

$$\text{Rule: IF} \begin{cases} \text{Port \# is of the Preferred Status} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.12. RequestURL(self) function

This function is used to check whether the external objects contained within a webpage such as images and videos, have been loaded from a different domain.

$$\text{Rule: IF} \begin{cases} \% \text{ of Request URL} < 22\% \rightarrow \text{Legitimate} \\ \% \text{ of Request URL} \geq 22\% \text{ and } 61\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{feature} = \text{Phishing} \end{cases}$$

5.1.13. AnchorURL(self) function

This function has been made to study the anchor tag of a given URL.

$$\underline{\text{Rule: IF}} \begin{cases} \% \text{ of URL Of Anchor} < 31\% \rightarrow \text{Legitimate} \\ \% \text{ of URL Of Anchor} \geq 31\% \text{ And } \leq 67\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

5.1.14. LinksInScriptTags(self) function

This function has been made to check whether <Meta>, <Script> and <Link> tags are common for legitimate websites, and it is expected that these tags are linked to the same domain of the web page.

Rule:

IF

$$\left\{ \begin{array}{l} \% \text{ of Links in " < Meta > ", " < Script > " and " < Link > " < 17\% \rightarrow \textit{Legitimate} \\ \% \text{ of Links in " < Meta > ", " < Script > " and " < Link > " \geq 17\% \text{ And } \leq 81\% \rightarrow \textit{Suspicious} \\ \text{Otherwise} \rightarrow \textit{Phishing} \end{array} \right.$$

5.1.15. ServerFormHandler(self) function

This function has been made to study the SFH of a given URL.

$$\text{Rule: IF} \left\{ \begin{array}{l} \text{SFH is "about: blank" Or Is Empty} \rightarrow \textit{Phishing} \\ \text{SFH Refers To A Different Domain} \rightarrow \textit{Suspicious} \\ \text{Otherwise} \rightarrow \textit{Legitimate} \end{array} \right.$$

5.1.16. AbnormalURL(self) function

This function has been made to check whether the host name is present in the URL.

$$\text{Rule: IF} \left\{ \begin{array}{l} \text{The Host Name Is Not Included In URL} \rightarrow \textit{Phishing} \\ \text{Otherwise} \rightarrow \textit{Legitimate} \end{array} \right.$$

5.1.17. WebsiteForwarding(self) function

This function is used to check the number of times the URL re directs after submitting it.

$$\text{Rule: IF} \left\{ \begin{array}{l} \text{ofRedirect Page} \leq 1 \rightarrow \textit{Legitimate} \\ \text{of Redirect Page} \geq 2 \text{ And } < 4 \rightarrow \textit{Suspicious} \\ \text{Otherwise} \rightarrow \textit{Phishing} \end{array} \right.$$

5.1.18. StatusBarCust(self) function

This function is used to check whether onMouseOver event changes the status bar.

$$\text{Rule: IF} \left\{ \begin{array}{l} \text{onMouseOver Changes Status Bar} \rightarrow \textit{Phishing} \\ \text{It Does't Change Status Bar} \rightarrow \textit{Legitimate} \end{array} \right.$$

5.1.19. DisableRightClick(self) function

This function is used to check whether right click event has been disabled for the website.

$$\text{Rule: IF} \begin{cases} \text{Right Click Disabled} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.20. UsingPopupWindow(self) function

This function is used to check whether the popup windows contain text fields.

$$\text{Rule: IF} \begin{cases} \text{Popup Window Contains Text Fields} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.21. IframeRedirection(self) function

This function is used to detect iframe attribute in the URL.

$$\text{Rule: IF} \begin{cases} \text{Using iframe} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.22. AgeofDomain(self) function

This function is used to extract the age of the domain of the website that belongs to the particular URL.

$$\text{Rule: IF} \begin{cases} \text{Age Of Domain} \geq 6 \text{ months} \rightarrow \text{Legitimate} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

5.1.23. DNSRecording(self) function

This function is used to check the DNS record for the domain related to the URL.

$$\text{Rule: IF} \begin{cases} \text{no DNS Record For The Domain} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.24. WebsiteTraffic(self) function

This function is used to check the traffic on the domain. Legitimate websites are ranked among the top 100,000. Furthermore, if the domain has no traffic or

is not recognized by the Alexa database, it is classified as "Phishing". Otherwise, it is classified as "Suspicious".

$$\text{Rule: IF} \begin{cases} \text{Website Rank} < 100,000 \rightarrow \text{Legitimate} \\ \text{Website Rank} > 100,000 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phish} \end{cases}$$

5.1.25. GoogleIndex(self) function

This function examines whether a website is in Google's index or not.

$$\text{Rule: IF} \begin{cases} \text{Webpage Indexed by Google} \rightarrow \text{Legitimate} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

5.1.26. LinksPointingToPage(self) function

This function is used to check the number of links pointing to the webpage of the given URL.

Rule:

IF

$$\begin{cases} \text{Of Link Pointing to The Webpage} = 0 \rightarrow \text{Phishing} \\ \text{Of Link Pointing to The Webpage} > 0 \text{ and } \leq 2 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.27. StatsReport(self) function

This function is used to find out whether the host of the website belongs to one of the top phishing IP's or not.

$$\text{Rule: IF} \begin{cases} \text{Host Belongs to Top Phishing IPs or Top Phishing Domains} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.28. PageRank(self) function

This parameter signifies how much importance of the webpage is on the internet this value is between 0 and 1 so if page rank less than 0.2 then website is Phishing and parameter value -1 else 1.

$$\text{Rule: IF} \begin{cases} \text{PageRank} < 0.2 \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$$

5.1.29. InfoEmail(self) function

This function is used to find whether mail() and mailto() events are present.

Rule: IF $\begin{cases} \text{Using "mail()" or "mailto:" Function to Submit User Information} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate} \end{cases}$

5.1.30. Https(self) function

This parameter signifies whether the site has added HTTPS in domain name to fool the target if yes then parameter value is -1 else 1.

5.2. Machine Learning Algorithms/ Techniques

Machine learning provides simplified and efficient methods for data analysis. It has indicated promising outcomes in Realtime classification problems recently. The key advantage of machine learning is the ability to create flexible models for specific tasks like phishing URL detection. Since phishing is a classification problem, Machine learning models can be used as a powerful tool. Machine learning models could adapt to changes quickly to identify patterns of fraudulent transactions that help to develop a learning-based identification system. Most of the machine learning models discussed here are classified as supervised machine learning, this is where an algorithm tries to learn a function that maps an input to an output based on example input-output pairs. It infers a function from labeled training data consisting of a set of training examples. We present machine learning methods that we used in our project.

5.2.1. Logistic Regression

Logistic Regression is a classification algorithm used to assign observations to a discrete set of classes. Unlike linear regression which outputs continuous number values, Logistic Regression transforms its output using the logistic sigmoid function to return a probability value which can then be mapped to two or more discrete classes. Logistic regression works well when the relationship in the data is almost linear despite if there are complex nonlinear relationships between variables, it has poor performance. Besides, it requires more statistical assumptions before using other techniques.

5.2.2. K-Nearest Neighbors

K-Nearest Neighbors (KNN) is one of the simplest algorithms used in machine learning for regression and classification problems which is non-parametric and lazy. In KNN there is no need for an assumption for the underlying data distribution. KNN algorithm uses feature similarity to predict the values of new datapoints which means that the new data point will be assigned a value based on how closely it matches the points in the training set. The similarity between records can be measured in many different ways. Once the neighbors are discovered, the summary prediction can be made by returning the most common outcome or taking the average. As such, KNN can be used for classification or regression problems. There is no model to speak of other than holding the entire training dataset.

5.2.3. Support Vector Machine

Support vector machines (SVMs) are one of the most popular classifiers. The idea behind SVM is to get the closest point between two classes by using the maximum distance between classes. This technique is a supervised learning model used for linear and nonlinear classification. Nonlinear classification is performed using a kernel function to map the input to a higher-dimensional feature space. Although SVMs are very powerful and are commonly used in classification, it has some weakness. They need high calculations to train data. Also, they are sensitive to noisy data and are therefore prone to overfitting. The four common kernel functions at the SVM are linear, RBF (radial basis function), sigmoid, and polynomial, which is listed in Table I. Each kernel function has particular parameters that must be optimized to obtain the best result.

Kernel Type	Formula	Parameter
Linear	$K(x_n, x_i) = (x_n, x_i)$	C, γ
RBF	$K(x_n, x_i) = \exp(-\gamma \ x_n - x_i\ ^2 + C)$	C, γ
Sigmoid	$K(x_n, x_i) = \tanh(\gamma(x_n, x_i) + r)$	C, γ, r
Polynomial	$K(x_n, x_i) = (\gamma(x_n, x_i) + r)^d$	C, γ, r, d

Table 1: Four common kernel functions

5.2.4. Naive Bayes Classifier

Naïve Bayes Classifier is a simple probabilistic classifier based on conditional probability. This classification algorithm uses basic Bayesian theorem and propagates a firm independence that is present between features. Occurrence of features along with their inter-relations is calculated in corpus consideration. It uses one class label. In NB, correlation of all neglected attributes is considered to be independent.

5.2.5. Decision Tree

Decision tree classifiers are used as a well-known classification technique. A decision tree is a flowchart-like tree structure where an internal node represents a feature or attribute, the branch represents a decision rule, and each leaf node represents the outcome. The topmost node in a decision tree is known as the root node. It learns to partition based on the attribute value. It partitions the tree in a recursive manner called recursive partitioning. This particular feature gives the tree classifier a higher resolution to deal with a variety of data sets, whether numerical or categorical data. Also, decision trees are ideal for dealing with nonlinear relationships between attributes and classes. Regularly, an impurity function is determined to assess the quality of the division for each node, and the Gini Variety Index is used as a known criterion for the total performance. In practice, the decision tree is flexible in the sense that it can easily model nonlinear or unconventional relationships. It can interpret the interaction between predictors. It can also be interpreted very well because of its binary structure. However, the decision tree has various drawbacks that tend to overuse data. Besides, updating a decision tree by new samples is difficult.

5.2.6. Random Forest

Random Forest, as its name implies, contains a large number of individual decision trees that act as a group to decide the output. Each tree in a random forest specifies the class prediction, and the result will be the most predicted class among the decision of trees. The reason for this amazing result from Random Forest is because of the trees protect each other from individual errors. Although some trees may predict the wrong answer, many other trees will rectify the final prediction, so as a group the trees can move in the right direction. Random Forests achieve a reduction in overfitting by combining many weak learners that underfit because

they only utilize a subset of all training samples Random Forests can handle a large number of variables in a data set. Also, during the forest construction process, they make an unbiased estimate of the generalization error. Besides, they can estimate the lost data well. The main drawback of Random Forests is the lack of reproducibility because the process of forest construction is random. Besides, it is difficult to interpret the final model and subsequent results, because it involves many independent decision trees.

5.2.7. Gradient Boosting Classifier

Gradient boosting is a machine learning technique used in regression and classification tasks, among others. It gives a prediction model in the form of an ensemble of weak prediction models, which are typically decision trees. When a decision tree is the weak learner, the resulting algorithm is called gradient-boosted trees; it usually outperforms random forest. A gradient-boosted trees model is built in a stage-wise fashion as in other boosting methods, but it generalizes the other methods by allowing optimization of an arbitrary differentiable loss function. The loss function is a measure indicating how good the models' coefficients are at fitting the underlying data. A logical understanding of loss function would depend on what we are trying to optimize.

5.2.8. CatBoost Classifier

CatBoost is an algorithm for gradient boosting on decision trees. It is a readymade classifier in scikit-learn's conventions terms that would deal with categorical features automatically. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today. It is used for search, recommendation systems, personal assistant, self-driving cars, weather prediction and many other tasks. Also, it provides best-in-class accuracy. It is powerful as it yields state-of-the-art results without extensive data training typically required by other machine learning methods, and provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems.

5.2.9. XGBoost Classifier

XG Boost stands for eXtreme Gradient Boosting. It is an application of gradient boosted decision trees, which is intended for its speed and performance. Boosting is an ensemble learning method where advanced techniques are included in order to rectify the errors made by the already proposed models. Models are included consecutively till we find that no additional enhancement can be carried out. While adding new models it uses a gradient descent technique to minimize the loss. The application of this algorithm is to provide efficient computational time and memory supplies. The aim of this design was to produce the best necessity of the accessible sources to train the model. Execution Speed and Model Performance are the two main reasons to work with XG Boost. This approach can support both classification and regression models.

5.2.10. Multi-layer Perceptron

The multilayer perceptron is the most widely used model of neural network. A significant part of the notoriety of MLPs can be owed to the way that they have been connected effectively to an extensive variety of data undertakings, including design grouping, workplace learning, and time arrangement forecast. Practical applications for MLPs have been found in such diverse fields as speech recognition, image compression, medical diagnosis, autonomous vehicle control, and financial prediction, and new applications are being discovered all the time. MLPs are trained, rather than programmed, to carry out the chosen information processing task. MLP training involves the adjustment of the network so that it can produce a specified output for each of a given set of input patterns. Since the desired outputs are known in advance, MLP training is an example of supervised learning. The MLP architecture consists units or nodes arranged in two or more layers (the input layer, which serves only to distribute the input from each pattern, is not counted). Real-valued weights connect some of the nodes, with no connections between nodes in the same layer.

5.3. Implementation Tools

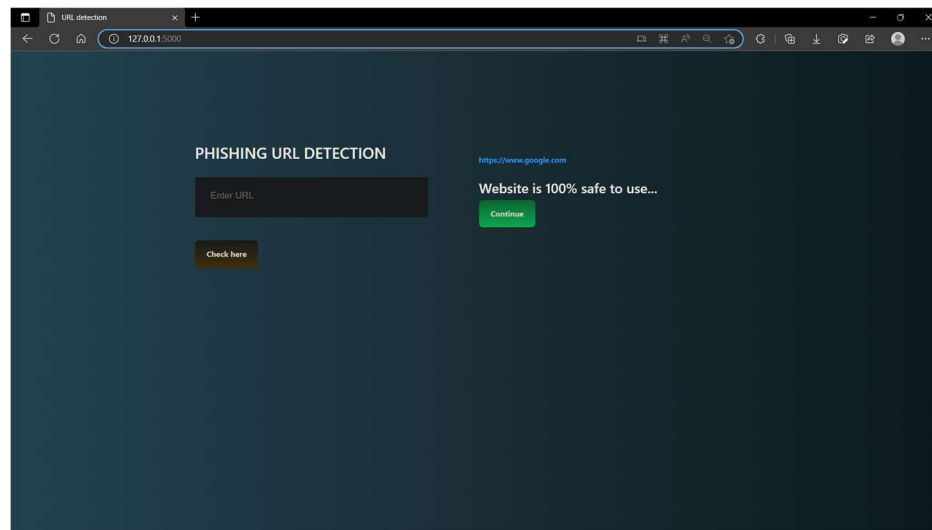
We use a laptop having a Core i7 processor with 2.60 GHz clock speed and 16 GB RAM to implement our proposed approach to detect phishing URLs. We use Python Programming Language due to its extensive support for libraries and short compile time. We use terminal to launch the website. We create different functions to extract necessary features. Some of the libraries used in the extraction process are: **re**: We use

this library to find the desired string from the URL; **urllib**: We use this library to get the response object from any URL and parse the URL components; **BeautifulSoup**: We use this useful library to extract information from XML and HTML documents and to create DOM (Document Object Model); **datetime**: This library is used because it supplies classes for manipulating dates and times; **googlesearch**: we use this library for searching Google, easily. googlesearch uses requests and BeautifulSoup4 to scrape Google; **dateutil.parser**: we use the parser module to parse datetime strings in many more formats.

5.4. Test cases

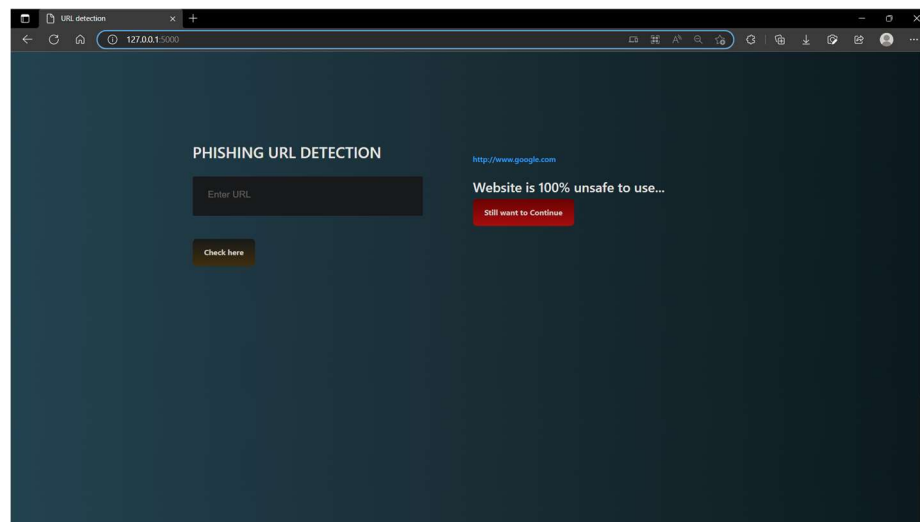
For the purposes of testing, we have decided to carry out several different test cases, each of which will include a mix of secure and vulnerable websites with different URL properties which might mark it as suspicious and more suitable to avoid. Because our project is only concerned with checking the URL of the website for signs that it might be a phishing website and is not actually concerned with verifying the contents of the website, the website checks the URL by examining the features it possesses and assigning it a rating based on the machine learning models. The machine learning models that determine the rating are taught the patterns that frequently appear in the URLs of phishing websites. The model with the highest accuracy rating is the one that we will use to determine whether or not the URL that we have entered leads to a phishing website. Simply executing the app.py file causes all of the other modules of our code to be called, which in turn deploys the website for us and allows us to launch the website. After that, all that is required of us is to enter the website's URL, and the results of the calculations, which are based on the URL, give an approximation of the percentage of the likelihood that the website in question is a phishing website. If the website is secure, a button will appear on the screen that will allow us to proceed to the website. If the website is not secure, a button will appear asking if we still want to proceed to the website anyway.

5.4.1 Testing <https://www.google.com> (https links)



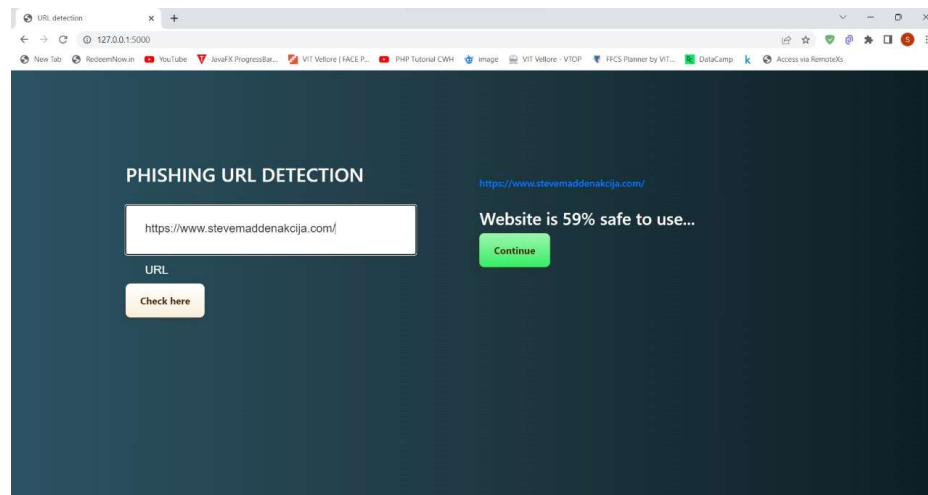
According to the ml model, the presence of the https communication protocol on a website is an indication that the website is trustworthy. This website also does not violate any of the other criteria that the model checks in order to determine if it is a phishing website; as a result, it has been given a rating that indicates that it is completely legitimate.

5.4.2 Testing <http://www.google.com> (http links)



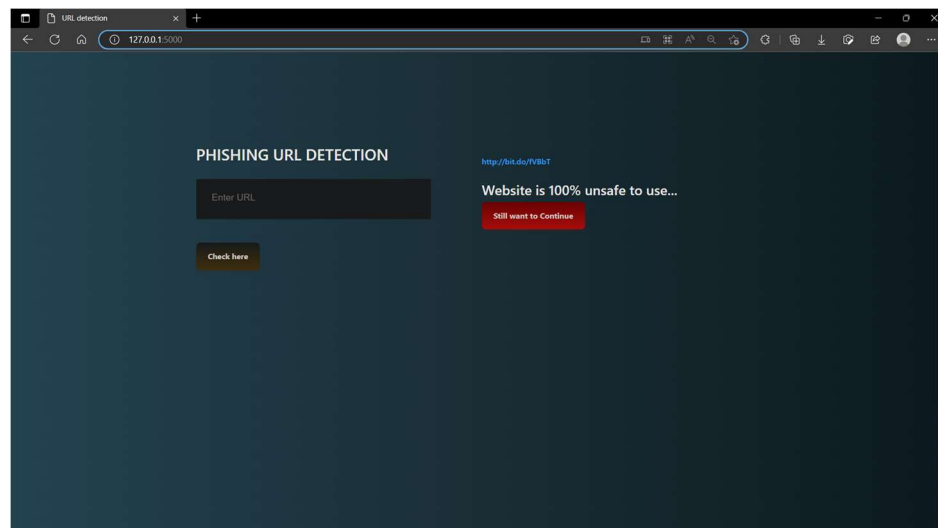
According to the ml model, the fact that this website employs the http communication protocol is indicative of the presence of a malicious website. When the communication protocol in and of itself is unsafe, the other criteria become irrelevant; as a result, this website has been given a rating that indicates that it is 100 percent unsafe.

5.4.3 Testing <https://www.stevemaddenakcija.com/> (Certificate expired)



The expiration date displayed in this record is the date the registrar's sponsorship of the domain name registration in the registry is currently set to expire. This date does not necessarily reflect the expiration date of the domain name registrant's agreement with the sponsoring registrar. As a result, the website has been classified as having a level of risk appropriate for moderate use.

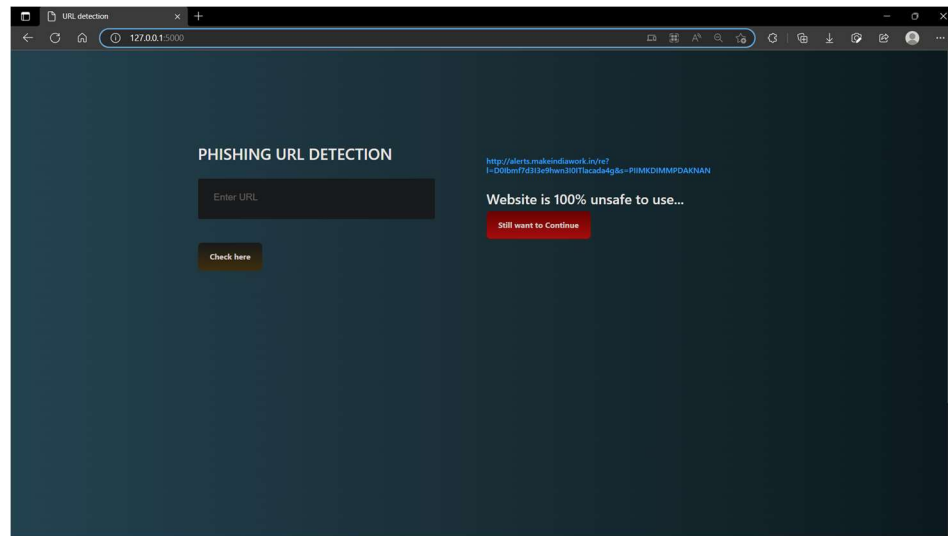
5.4.4 Testing <http://bit.do/fVBbT> (Short links)



The ml model has reached the conclusion that certain short URLs are dangerous because they have the potential to lead us to websites that are unsafe. Because it is impossible to evaluate the website that will be generated from the shortened URL, it is strongly recommended that you stay away from it, and as a result, it has been given a rating of "100 percent unsafe."

5.4.5 Testing

<http://alerts.makeindiaaork.in/re?l=D0lbmf7d3I3e9hwn3I0ITlacada4g&s=PIIMKDIMMPDAKNANn> (Long links)

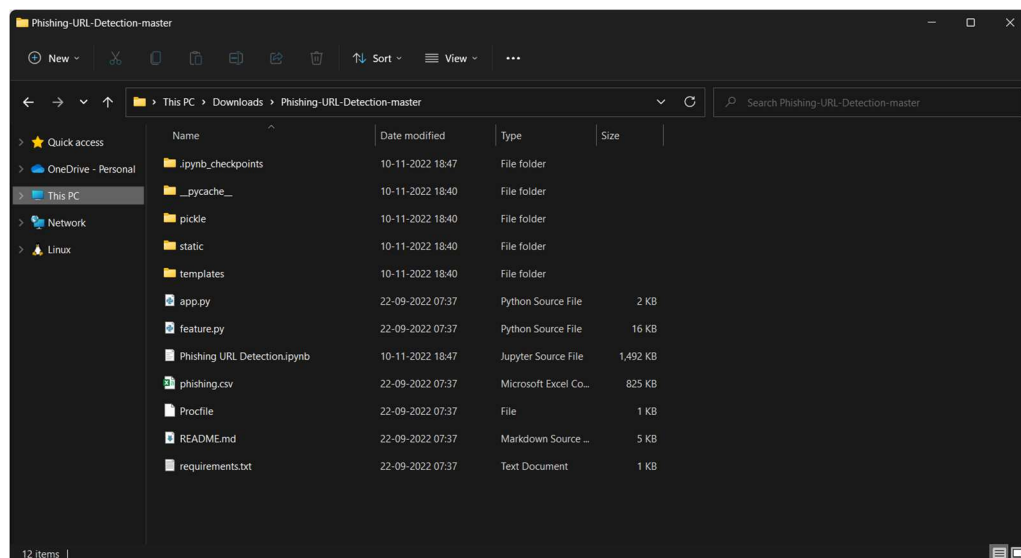


Extremely lengthy links are another indication that a website may contain potentially harmful content. This pattern can be found on a number of websites that are used for phishing. Because of this, the ML model has determined that the website in question is risky and recommends that users stay away from it.

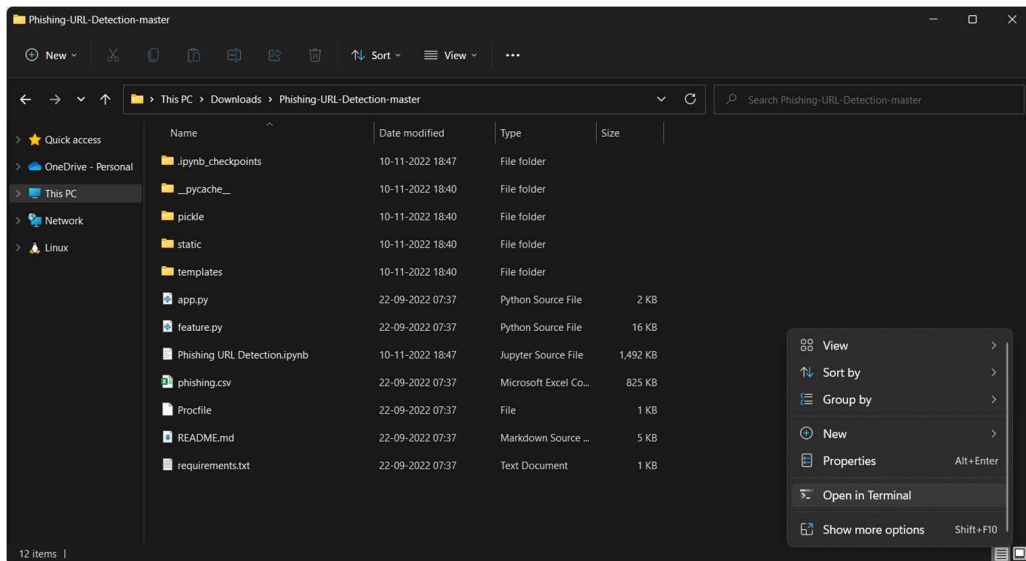
5.5. Execution of the project

The website that checks whether the given link entered is associated with a phishing website or is a legitimate website can be executed in a relatively straightforward manner.

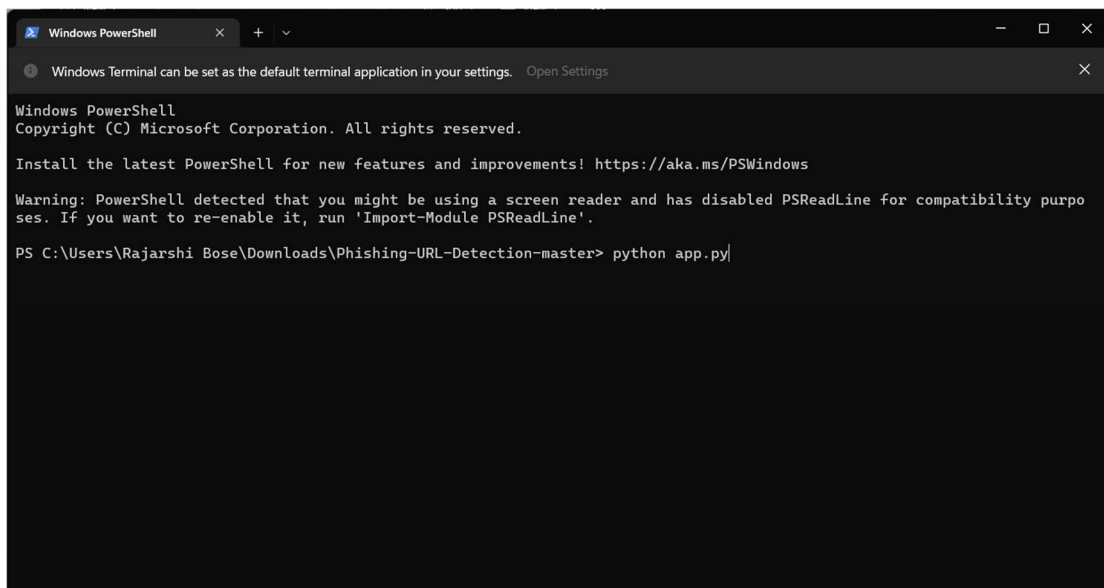
1) To begin, we will need to open the directory that contains the project files.



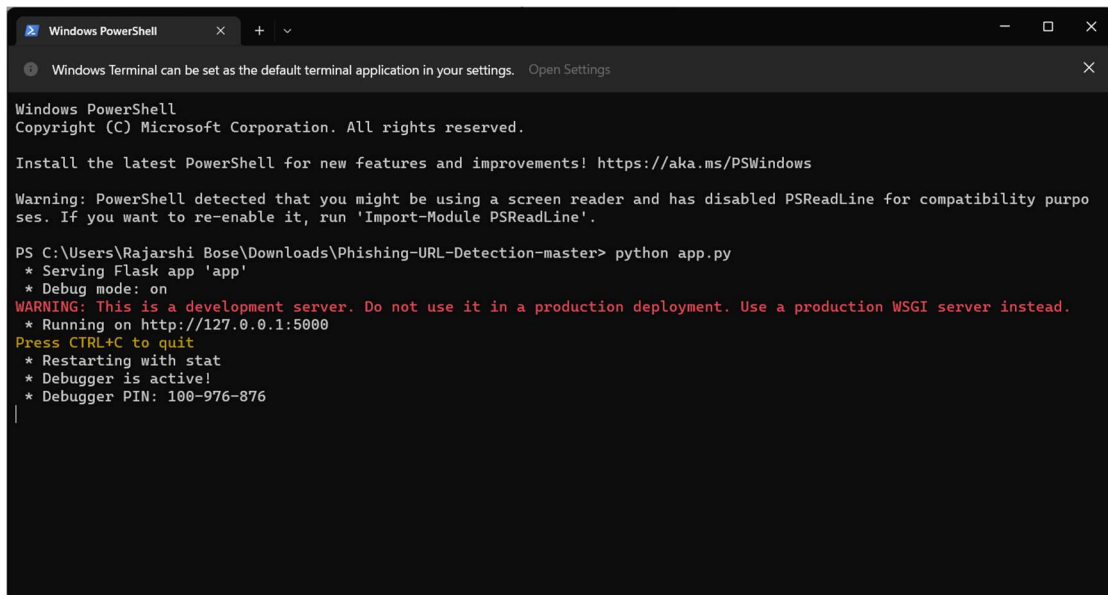
2) Select open in terminal from the context menu when you right-click anywhere. This makes it possible for the terminal to start up with the directory of the project selected.



3) Start the Python project by running the app.py file with a basic command called python app.py. This will allow the project to be run.



4) The code will be put into action, and the website will be hosted on a deployment server, which in most cases can be found at <http://127.0.0.1:5000>.



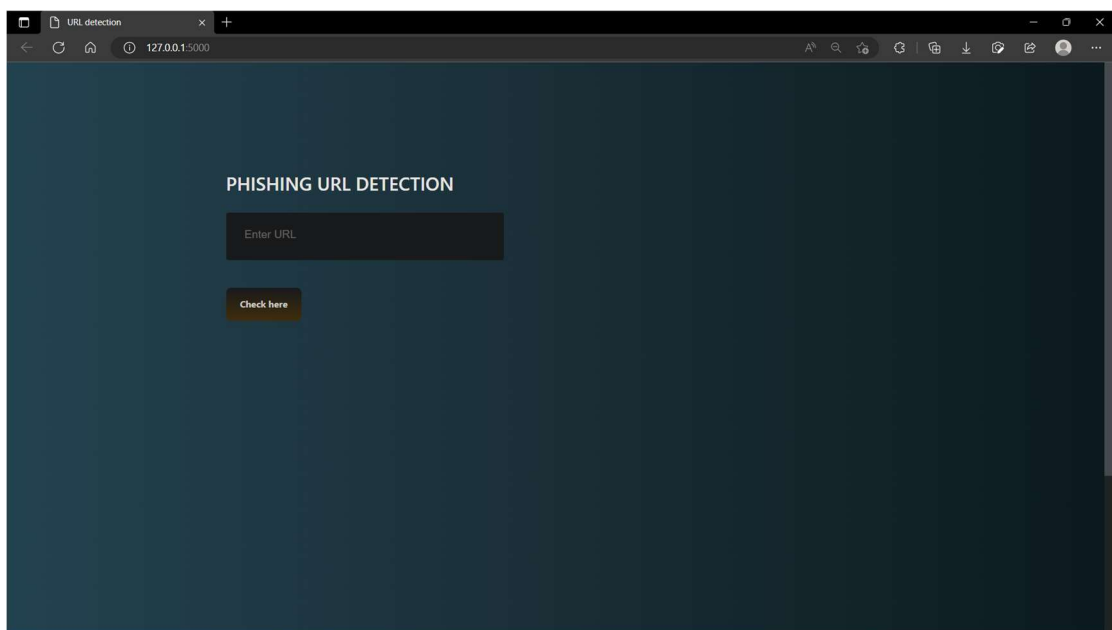
```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

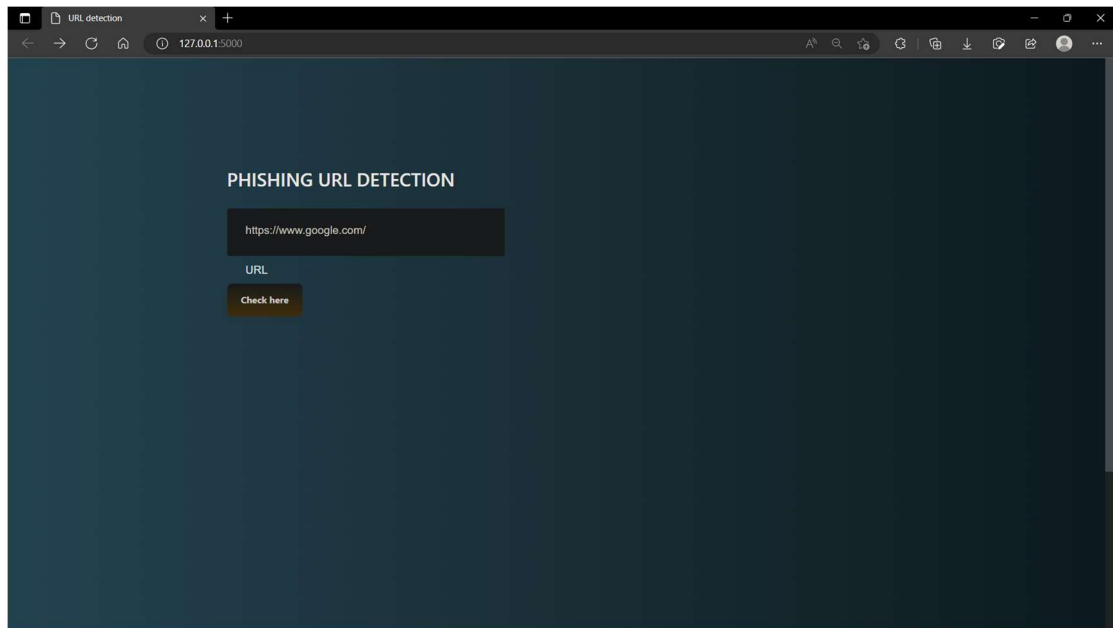
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Users\Rajarshi Bose\Downloads\Phishing-URL-Detection-master> python app.py
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 100-976-876
```

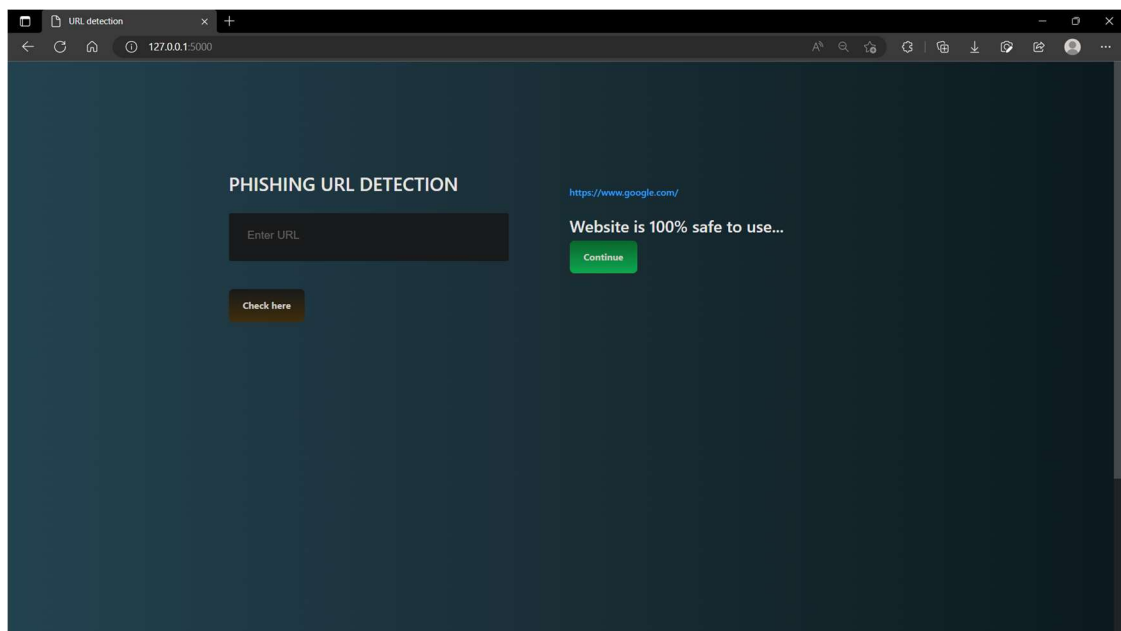
5) If we click on the link, we will be taken to the website in question.

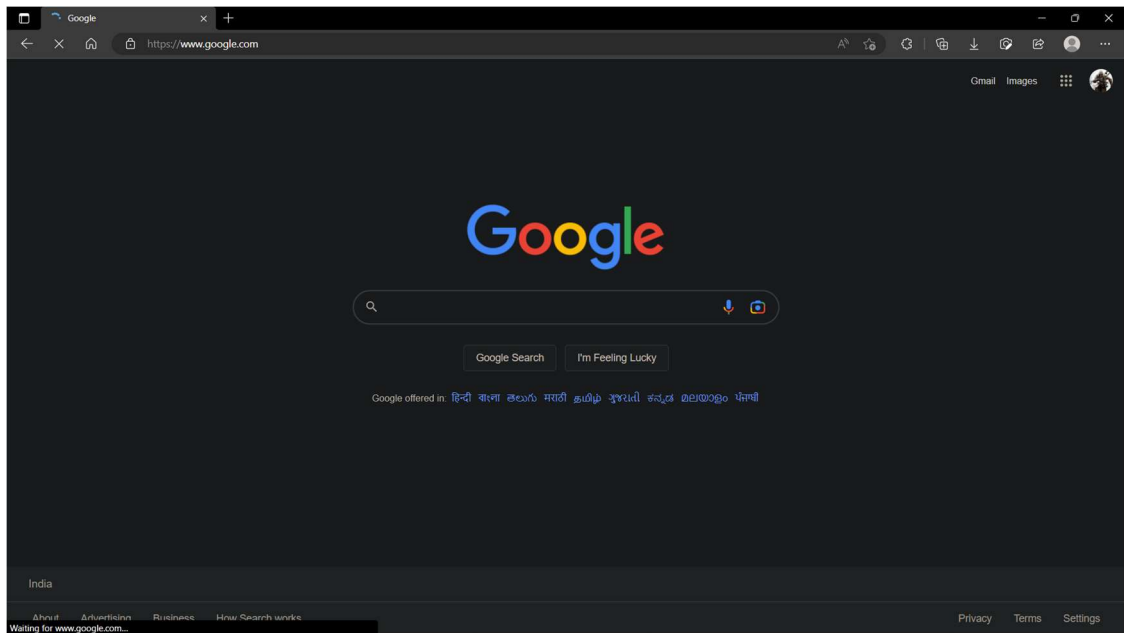


6) At this point, all we need to do is enter the link to the website that needs to be tested and then click the "Check Here" button.



7) After this step, which will take some time, the results will be displayed as a percentage indicating the level of danger posed by the URL you entered, and you will be presented with a button that will take you to the website you entered.





5.6. Results analysis

	ML Model	Accuracy	f1_score	Recall	Precision
0	Gradient Boosting Classifier	0.974	0.977	0.994	0.986
1	CatBoost Classifier	0.972	0.975	0.994	0.989
2	Random Forest	0.971	0.974	0.993	0.990
3	Support Vector Machine	0.964	0.968	0.980	0.965
4	Decision Tree	0.960	0.964	0.991	0.993
5	K-Nearest Neighbors	0.956	0.961	0.991	0.989
6	Logistic Regression	0.934	0.941	0.943	0.927
7	Naive Bayes Classifier	0.605	0.454	0.292	0.997
8	Multi-layer Perceptron	0.549	0.549	0.988	0.988
9	XGBoost Classifier	0.540	0.540	0.952	0.952

Figure 4: Machine Learning models with their evaluation parameters (Table sorted in descending order based on their accuracy and f1_score)

In Figure 4 shows the ML models used in the URL analysis, sorted in descending order based in accuracy and f1_score. Gradient boost shows the most success with the accuracy of 0.974, with XGBoost Classifier being the least successful with the accuracy of 0.540. In terms of f1_score, Gradient boost and XGBoost Classifier are once again the best and worst

models respectively. In Recall, both Gradient boost and CatBoost are the best with Naïve Bayes performing the least. The most precise model was Naïve Bayes with 0.997 and the least precise was Logistic Regression with 0.927.

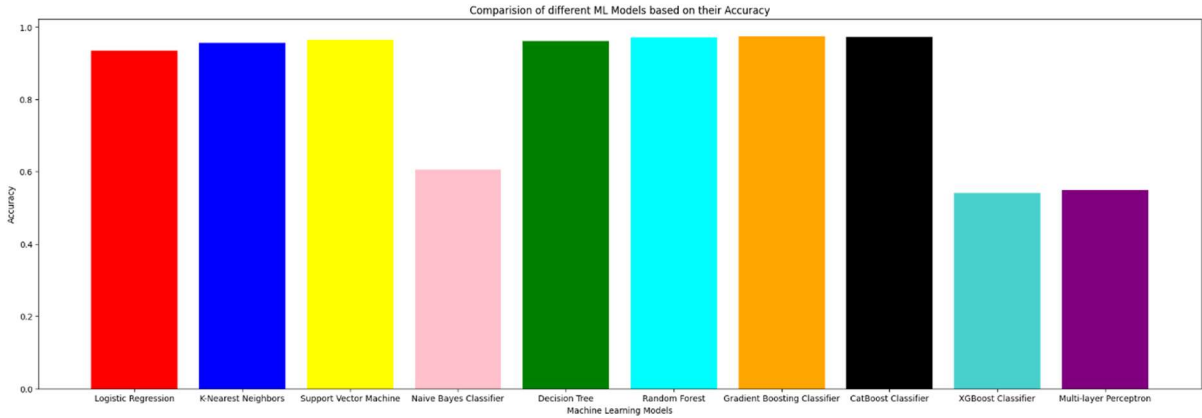


Figure 5: Bar plot showing the comparison of different ML models based on their accuracy

In Figure 5 we can clearly see a bar plot comparing different ML models based on their accuracy with Gradient boosting classifier having a slight advantage.

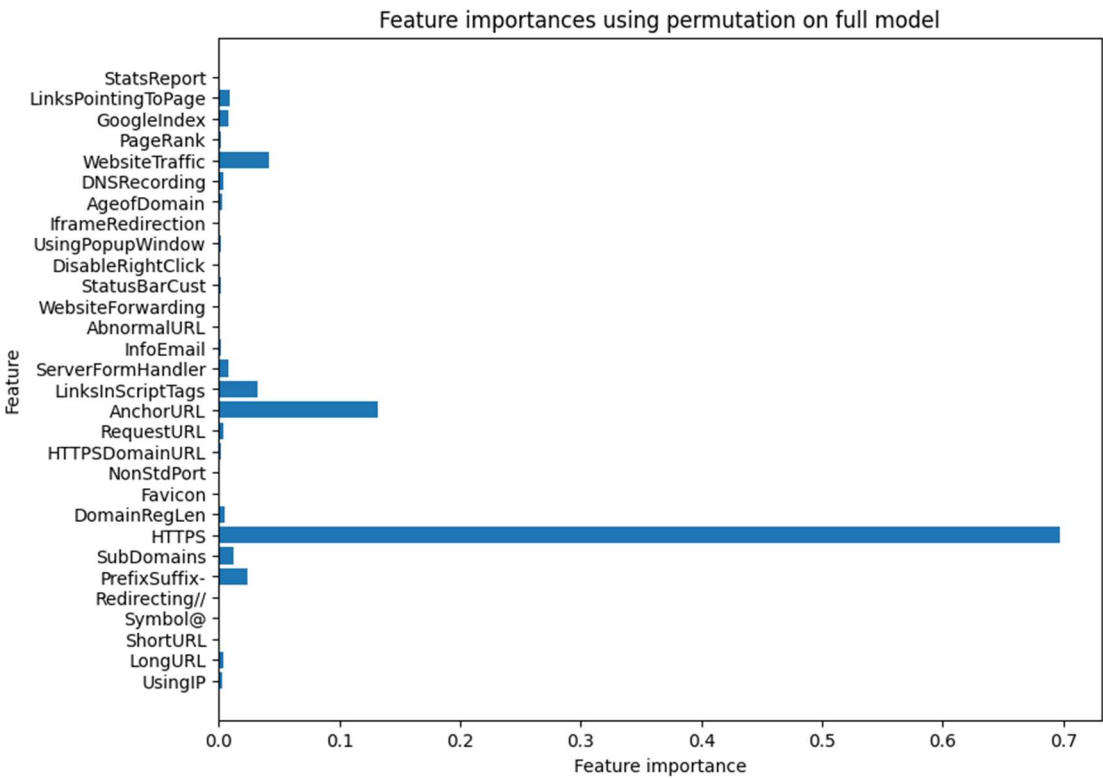


Figure 6: Bar graph showing feature importance using permutation on full model

In figure 6 the bar graph shows the importance of each of the 30 features when deciding the likelihood of a URL's legitimacy. Possessing HTTPS protocol is by far the most reliable and most important factor in determining if a URL is safe. An Anchor URL is the second most important feature, with the nature of the website traffic being the most important after that. Certain features such as possessing a short url and website forwarding have very little relative importance.

Gradient boost is the best performing algorithm for several reasons. Gradient boost works by generating a strong learner from an additive model of weak learners. Due to the way Gradient boost trains on the errors of the strong learner instead of modifying the sample distribution, the pseudo residuals are used to fit the weak learner to them. This results in a relatively high accuracy for gradient boost when compared to other algorithms, across many problems.

6. Conclusion and Future Scope

It is remarkable that we are able to make predictions using machine learning about the characteristics of a phishing URL and then use those predictions to make predictions about the likelihood of a website being a potential phishing URL. Because of this project, we gained a lot of knowledge about a wide range of different aspects of machine learning, such as the ten primary machine learning models and the way the algorithms investigate the dataset. In addition to this, we were able to gain an understanding of the qualities of a URL that determine the likelihood of it being unsafe. However, the accuracy of the machine learning model needs to be improved over time by continually retraining it with datasets that are both larger and more detailed. In order to identify possible phishing websites, we intend to use this as a foundation to also incorporate convolutional neural networks (CNN) into our project. The approach that has been proposed makes use of CNN for high-precision analysis in order to differentiate between authentic destinations and phishing locales. CNN-based models end up being exceptionally useful in identifying obscure phishing websites as a direct result of the findings of extensive studies. In addition, the CNN-based methodology is typically carried out in a manner that is superior to the conventional ML classifiers that are evaluated on the same dataset. As a result, it achieves a phishing location rate of 98.2% and an F1-score of 0.976. This is the path that our project will be heading in the future in order to make this system more reliable and precise in its detection of phishing websites.

7. References

1. Chawla, A. (2022, March). Phishing website analysis and detection using Machine Learning. *International Journal of Intelligent Systems and Applications in Engineering (IJISAE)*, 10(1), 10–16. DOI: <https://doi.org/10.18201/ijisae.2022.262>
2. Kumar, j., Santhanavijayan, A., Janet, B., Rajendran, B. & Bindhumadhava, B. S. (2020). Phishing Website Classification and Detection Using Machine Learning. *International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1-6, DOI: 10.1109/ICCCI48352.2020.9104161
3. Mahajan, R. & Siddavatam, I. (2018, October). Phishing Website Detection using Machine Learning Algorithms. *International Journal of Computer Applications* (0975 – 8887), Volume 181 – No. 23. DOI: 10.5120/ijca2018918026
4. Dutta, A. K. (2021, October). Detecting phishing websites using machine learning technique. *PLoS ONE* 16(10): e0258361. DOI: <https://doi.org/10.1371/journal.pone.0258361>
5. Sampat, H., Saharkar, M., Pandey, A. & Lopes, H. (2018, March). Detection of Phishing Website Using Machine Learning. *International Research Journal of Engineering and Technology (IRJET)*, Volume: 05 Issue: 03. DOI: <https://irjet.net/archives/V5/i3/IRJET-V5I3580.pdf>
6. Deshpande, A., Pedamkar, O., Chaudhary, N. & Borde, S. (2021, May). Detection of Phishing Websites using Machine Learning. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 10 Issue 05. DOI: 10.17577/IJERTV10IS050235
7. Pratik, N. N., Vaneeta, M., Prajwal, D., Pradeep, K. S. & Kakade, S. K. (2020, June). Detection of Phishing Websites Using Machine Learning Techniques. *International Journal of Emerging Technologies and Innovative Research (JETIR)*, Vol.7, Issue 6, page no.117-123. DOI: <http://www.jetir.org/papers/JETIR2006018.pdf>
8. Patil, V., Thakkar, P., Shah, C., Bhat, T. & Godse, S.P. (2018). Detection and Prevention of Phishing Websites Using Machine Learning Approach. *Fourth International Conference on Computing Communication Control and Automation (ICCUBEA)*, pp. 1-5. DOI: 10.1109/ICCUBEA.2018.8697412
9. Garcés, I. O., Cazares, M. F. & Andrade, R. O. (2019). Detection of Phishing Attacks with Machine Learning Techniques in Cognitive Security Architecture. *International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 366-370. DOI: 10.1109/CSCI49370.2019.00071

10. Niakanlahiji, A., Chu, B. -T. & Al-Shaer, E. (2018). PhishMon: A Machine Learning Framework for Detecting Phishing Webpages. IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 220-225. DOI: 10.1109/ISI.2018.8587410
11. Alswailem, A., Alabdullah, B., Alrumayh, N. & Alsedrani, A. (2019). Detecting Phishing Websites Using Machine Learning. International Conference on Computer Applications & Information Security (ICCAIS), pp. 1-6. DOI: 10.1109/CAIS.2019.8769571
12. Abdelhamid, N., Thabtah, F. & Abdel-jaber, H. (2017). Phishing detection: A recent intelligent machine learning comparison based on models content and features. IEEE International Conference on Intelligence and Security Informatics (ISI), pp. 72-77. DOI: 10.1109/ISI.2017.8004877
13. Das Gupta, S., Shahriar, K.T., Alqahtani, H. et al (2022, March). Modeling Hybrid Feature-Based Phishing Websites Detection Using Machine Learning Techniques. Ann. Data. Sci. DOI: <https://doi.org/10.1007/s40745-022-00379-8>
14. Chatterjee, M. & Namin, A. -S. (2019). Detecting Phishing Websites through Deep Reinforcement Learning. IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), pp. 227-232. DOI: 10.1109/COMPSAC.2019.10211
15. Selvakumari, M & Sowjanya, M & Das, Sneha & Padmavathi, S. (2021). Phishing website detection using machine learning and deep learning techniques. Journal of Physics: Conference Series. 1916. 012169. DOI: 10.1088/1742-6596/1916/1/012169.
16. Mamun, M. S. I., Rathore, M. A., Lashkari, A. H., Stakhanova, N., & Ghorbani, A. A. (2016, September). Detecting malicious urls using lexical analysis. In International Conference on Network and System Security (pp. 467-482). Springer, Cham.
17. Zamir, A., Khan, H.U., Iqbal, T., Yousaf, N., Aslam, F., Anjum, A. and Hamdani, M. (2020). Phishing web site detection using diverse machine learning algorithms. The Electronic Library, Vol. 38 No. 1, pp. 65-80. DOI: <https://doi.org/10.1108/EL-05-2019-0118>
18. Kulkarni, A. & Brown III, L. L. (2019). Phishing Websites Detection using Machine Learning. International Journal of Advanced Computer Science and Applications (IJACSA), 10(7). DOI: <http://dx.doi.org/10.14569/IJACSA.2019.0100702>
19. Patil, R. R., Kaur, G., Jain, H., Tiwari, A., Joshi, S., Rao, K. & Sharma, A. (2022). Machine learning approach for phishing website detection: A literature survey. Journal

of Discrete Mathematical Sciences and Cryptography, 25:3, 817-827. DOI: 10.1080/09720529.2021.2016224

20. Odeh, A., Keshta, I. & Abdelfattah, E. (2021). PhiBoost- A novel phishing detection model Using Adaptive Boosting approach. Jordanian Journal of Computers and Information Technology (JJCIT), 07(01):65-74. DOI: 10.5455/jjcit.71-1600061738