REAL-TIME PHISHING DETECTION SYSTEM USING MACHINE LEARNING

ST. TERESA’S COLLEGE (AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

By

GIFTY BRIJIT T J - SB21BCA016

&

HARSHIKA C - SB21BCA017

III DC BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

Under the guidance of

Ms. SREELAKSHMY I J

DEPARTMENT OF BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

MARCH 2024

REAL-TIME PHISHING DETECTION SYSTEM USING MACHINE LEARNING

ST. TERESA'S COLLEGE (AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

By

GIFTY BRIJIT T J - SB21BCA016

&

HARSHIKA C - SB21BCA017

III DC BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

Under the guidance of

Ms SREELAKSHMY I J

DEPARTMENT OF BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

MARCH 2024

DECLARATION

We, undersigned, hereby declare that the project report, **“REAL-TIME PHISHING DETECTION SYSTEM USING MACHINE LEARNING”**, submitted for partial fulfillment of the requirements for the award of degree of BCA (Cloud Technology and Information Security Management)at St. Teresa’s College (Autonomous), Ernakulam (Affiliated to Mahatma Gandhi University), Kerala, is a bonafide work done by us under the supervision of Ms. Sreelakshmy I J. This submission represents our ideas in our own words and where ideas or words of others have not been included. We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Ernakulam
March 2021

Gifty Brijit T J – SB21BCA016
Harshika C – SB21BCA017

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

DEPARTMENT OF BCA (CLOUD TECHNOLOGY AND INFORMATION SECURITY MANAGEMENT)

CERTIFICATE

This is to certify that the report entitled "**REAL-TIME PHISHING DETECTION SYSTEM USING MACHINE LEARNING**", submitted by Gifty Brijit T J and Harshika C to the Mahatma Gandhi University in partial fulfillment of the requirements for the award of the Degree of BCA (Cloud Technology and Information Security Management) is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

ST. TERESA'S COLLEGE
DEPT. OF COMPUTERS
ERNAKULAM

Head of the department

20/03/2024

Internal Supervisor

26/mar/24

External Supervisor

ACKNOWLEDGEMENT

First and foremost we thank God Almighty for his blessings. We take this opportunity to express our gratitude to all those who helped us in completing this project successfully. I wish to express our sincere gratitude to the **Manager Rev. Dr. Sr. Vinitha CSST** and the Principal **Dr. Alphonsa Vijaya Joseph** for providing all the facilities.

We express our sincere gratitude towards the Head of the department
for the support. We deeply express sincere thanks to our project guide **Ms. Sreelakshmy I J** for her proper guidance and support throughout the project work.

We are indebted to our beloved teachers whose cooperation and suggestion throughout the project which helped us a lot. We thank all our friends and classmates for their support.

We convey our hearty thanks to our parents for the moral support, suggestion and encouragement.

ABSTRACT

Website phishing is the biggest threat in today's cyber security world. This is because the attackers are using advanced technologies and are becoming advanced day by day. A phishing attacker fakes a legitimate website such as that of a bank and invites the user to share credentials such as password and username and finally steal the money of the user. The fake one will have the complete features of the original website including color theme, logo, texts, and appearance so distinguishing the fake one and legitimate one will be challenging. There have been many technologies to track and detect a phishing website such as signature-based detection. Phishing can be detected in many ways and using many techniques. URL-based Phishing website detection using Machine Learning (ML) and Deep Learning(DL) is one of the most accurate techniques among them. This project develops a Chrome browser extension to detect phishing websites in real-time with a well-trained machine learning model at its back-end. The technique is used in URL-based phishing detection and so a huge amount of phishing URLs and legitimate URLs are collected from various sources and repositories. These URLs are processed in the feature extraction stage to extract around 30 features of the URL such as domain name, domain age, etc. This processed URL data is used to train ML algorithms such as K-Nearest Neighbor, Random Forest Classifier, and Decision Tree. The best performing ML model is finalized based on testing and evaluating on the basis of performance evaluation metrics. And then the best performing model is deployed as a Google Chrome extension in which the front-end has been developed in HTML. The extension will monitor all the websites loaded using the ML model at the back-end and give real-time alerts to the front-end. The machine learning development will be in Python 3.7 with supporting libraries such as numpy, pandas, sci-kit learn, etc in Google Colab software.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

DL - Deep Learning

ML - Machine Learning

CNN - Convolutional Neural Network

XGBOOST - eXtreme Gradient Boost

HTTP - Hypertext Transfer Protocol

HTML - Hyper Text Markup Language

MTM - Man in The Middle

URL - Uniform Resource Locator

CNN - Convolutional Neural Network

RF - Random Forest

Chapter 1

INTRODUCTION

Identity theft can be a crime where the perpetrator sends a false e-mail, or URL of a website that appears to come from a legitimate source or credible organization, requesting a personal certificate such as bank ownership, username, number, address, capital card details, and more. Fraudulent emails and websites often look strangely legitimate, and even a website whenever a net user is asked to enter personal data, and it sounds fair. Phishing scams are circulating via e-mail, SMS, instant messengers, social networking sites, VoIP, etc., however e-mail that widespread gratitude for these attacks and phishing scams is achieved by visiting the e-mail link. In addition, the criminal attacks of identity theft are changing dramatically these days.

The crime of stealing sensitive information still poses a severe threat to security, and a large number of internet users fall victim to this scam. Moreover, such attacks do not only cause problems for internet users but also for companies that provide online financial services. That is because when users fall victim to such a crime of identity theft, an online service provider often loses its reputation and economic damage.

1.1 Phishing

Phishing costs Internet users billions of dollars per year. It refers to luring techniques used by identity thieves to fish for personal information in a pond of unsuspecting Internet users. Phishers use spoofed email, phishing software to steal personal information and financial account details such as usernames and passwords. Social engineering schemes use spoofed emails, purporting to be from legitimate businesses and agencies, designed to lead consumers to counterfeit websites that trick recipients into divulging financial data such as usernames and passwords.Technical subterfuge schemes install malicious software onto computers, to steal credentials directly, often using systems to intercept consumers' online account usernames and passwords.

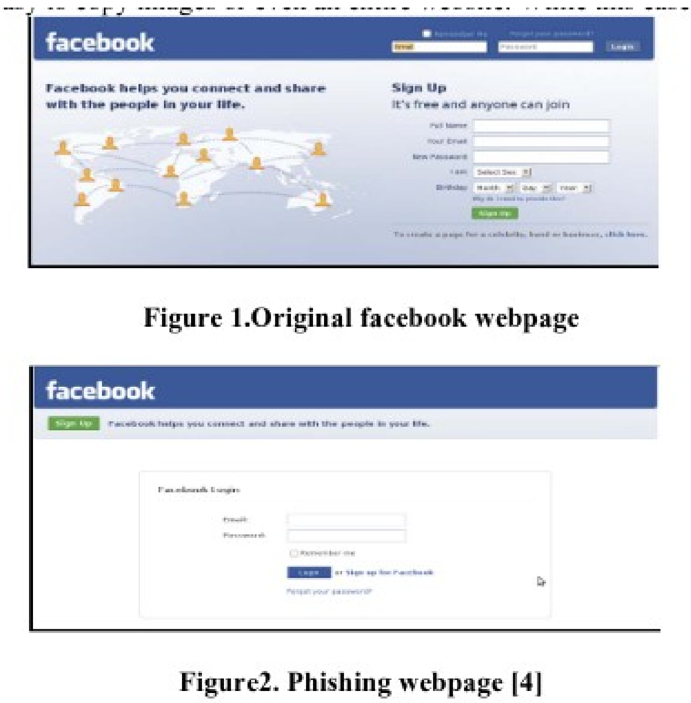

facebook
Facebook helps you connect and share with the people in your life.
Sign Up
It's free and anyone can join
facebook
Sign Up
Facebook helps you connect and share with the people in your life.
Facebook Login

Figure 1.Original facebook webpage

Figure2. Phishing webpage [4]

Figure 1.1: Example of phishing website

Figure 1.1 represents the webpage of the popular website www.facebook.com and its phish. The second image here represents a webpage similar to that of facebook, but is the webpage of a site which spreads phishing activities. A user may misunderstand the second site as a genuine facebook site and provide his personal identity details. The Phisher can thus steal that information and he may use it for vicious purposes.

A cybercrime website hits online businesses, banks, Web users, and governments, so it has become a national security issue. It is necessary that this attack be detected early. However, it is difficult to see these attacks because of the new methods used by criminals to steal sensitive information to commit crimes. For successful criminal detection of identity theft to be achieved, it must be obtained with the highest accuracy and in the shortest possible time. The most common method of detecting identity theft involves black listing and white listing.

Criminals use URLs for stealing sensitive information can be obtained with the concept of machine learning, which can be used continuously to prevent such attacks. First, machines used to follow instructions given to man, but now people can train a machine to learn from previous data, build a prediction model and perform very fast, and this is known as machine learning. It is basically the use of tools and technologies that can be used effectively and efficiently. Machine learning is used to make one's work easier, faster, and more accessible by learning from past data and working efficiently now. Machine language uses data to answer questions. This data usage is called the Training process, which uses our data to create and transform it into a predictable model. We may provide data on our machine, and it will assist in identifying and locating sensitive web theft websites.

A look at the history of phishing reveals that the first phishing email is thought to have originated sometime around the year 1995. The first many knew of the existence of phishing was five years later when the Love Bug struck. Fast forward almost twenty years and phishing is the number one attack vector for compromising an organization and stealing data. How did we get to this point? When did the bad guys get so savvy? Maybe there are some clues in the history of phishing.

Back in the early to mid-1990s, the only Internet option was ‘dial-up’ access for a fee. For those that were reluctant to pay for Internet access, the alternative was a thirty days free trial to access to the Internet via an AOL floppy disk. Rather than face life without the Internet after the trial period expired, some found a way to change their screen names to make it appear as if they were AOL administrators. Using these phony screen names, they would “phish” for log-in credentials to continue accessing the Internet for free. As Internet use increased in popularity, scammers adapted these tactics to disguise themselves as administrators from an ISP, emailing the accounts of the ISP’s customers to elicit user login credentials. Having spoofed someone, the hacker could access the Internet from that user’s account with the bonus of sending spam from the user’s email address.

The Love Bug of 2000

A change in tactics saw the world fall victim to the Love Bug on May 4 2000. Starting in the Philippines, mailboxes around the globe were filled with a message titled “ILOVEYOU”. The message body simply said “Kindly check the attached LOVELETTER coming from me”. Those who could not resist unearthing their secret crush, opened what they thought was a harmless .txt file, only to unleash a worm that did damage on the local machine. The worm overwrote image files and sent a copy of itself to all the user´s contacts in their Outlook address book. 'LoveBug' showed how to get spam to send itself and that, with a cleverly designed virus that preyed on human psychology and technical failings, malware could rack up enormous numbers of victims. In all about 45 million Windows PCs were thought to have been hit. The history of phishing shows that, although delivery methods have evolved over two decades to evade detection by spam filters and other technology, the tactics employed by phishers have remained fairly consistent. It would seem logical that people should have learned to avoid the trap of surrendering login credentials, clicking links or even opening attachments.

1.2 How Does Phishing Work?

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 email 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 sends the"spoofed" emails to as many people as possible in an attempt to lure them into the scheme. When these emails are opened when a link in the mail is clicked, the consumers are redirected to a spoofed website, appearing to be from the legitimate entity.

1.3 Statistics Of Phishing Attacks.

Phishing continues to be one of the rapidly growing classes of identity theft scams on the internet that is causing both short term and long term economic damage. There have been nearly 33,000 phishing attacks globally each month in the year 2012,totalling a loss of $687 million [1]. An example of phishing occurred in June 2004. The RoyalBank of Canada notified customers that fraudulent emails purporting to originate from the Royal Bank were being sent out asking customers to verify account numbers and personal identification numbers (PINs) through a link included in the email. The fraudulent email stated that if the receiver did not click on the link and key in his client card number and passcode, access to his account would be blocked. These emails were sent within a week of a computer malfunction that prevented customer accounts from being updated [2].The United States continued to be the top country hosting phishing sites during the third quarter of 2012. This is mainly due to the fact that a large percentage of the world's websites and domain names are hosted in the United States. FinancialServices remains to be the most targeted industry sector by phishers.

This project investigates and develops methods to detect phishing URLs using Machine Learning and Deep Learning techniques. A number of machine learning and deep learning models are trained to detect phishing URLs from unknown URLs. The models for phishing URL detection is trained on well-prepared data collected using web scraping. The phishing and legitimate URLs are collected from various sources such as www.phishtank.com and www.kaggle.com. This URL list is passed through a well-engineered preprocessing stage where around 30 features are extracted and preprocessed. This efficient dataset is used to train the models. The important stages are the feature extraction and preprocessing. Machine Learning models such as KNN (K-Nearest Neighbour), Decision Tree, Random Forest, are trained. They are evaluated based on evaluation parameters such as accuracy and precision. The trained models for phishing detection are saved as pickle files. Then a final app has been developed as a browser extension for Google Chrome.

Types of phishing:

There are two primary types of attacks.

Standard attacks: This method targets a large number of individuals and counts on one or more victims. The attacker understands that this approach is scattershot. However, that isn't of much consequence since the attacker only needs one successful victim to gain a foothold.

These scams target a wide audience with general bait.

Example of a standard attack

1. An attacker sends a mass email to employees posing as a member of the IT department.
2. The email is a notification for recipients to take the mandatory annual online IT security training module—however, the training module is attacker controlled.
3. During the course, the victim user is directed to enter their employee credentials which are then delivered directly to the attacker. A mass distribution is also a double-edged sword. The potential for luring in at least one victim is higher with a larger distribution. At the same time, the likelihood of gaining the attention of the organization's real IT or security teams is also higher.

Spear phishing. Compared to standard strategies, this is a more targeted attack. It requires more time and effort on behalf of the attacker since it targets fewer individuals through a carefully manipulated email. It's also common for the attacker to spend time building trust with the target before directing them to take malicious actions. This type of attack is more commonly used to place malware on an internal network.

Example of a spear phishing attack

1. An attacker becomes aware of a sensitive internal project at a target organization.
2. The attacker spoofs the original sender's email address.
3. The attacker sends out an otherwise innocuous email to the limited recipient list with the subject line, "Minutes from the last meeting" or "Action Items."
4. The recipients see what looks to be a legitimate email about a recent meeting regarding the project. Because there's an implicit trust, they are much more likely to open the attachment.

Such campaigns have been used to gain access to internal networks used by high-level executives in an organization who are authorized to access more sensitive information. The result is the same as a general operation, except the compromise occurs much deeper within the organization. Spear phishing aims to extract specific information or gain specific access to an internal network.

Example of Spear Phishing

An attacker tried to target an employee of NTL World, which is a part of the Virgin Media company, using spear phishing. The attacker claimed that the victim needed to sign a new employee handbook. This was designed to lure them into clicking a link where they would have been asked to submit private information.

Whaling

Whaling is an even more targeted type of phishing that goes after the whales – a marine animal even bigger than a fish. These attacks typically target a CEO, CFO, or any CXX within an industry or a specific business. A whaling email might state that the company is facing legal consequences and that you need to click on the link to get more information. The link takes you to a page where you are asked to enter critical data about the company such as tax ID and bank account numbers.

Smishing

Smishing is an attack that uses text messaging or short message service (SMS) to execute the attack. A common phishing technique is to deliver a message to a mobile phone through SMS that contains a clickable link or a return phone number. A common example of a smishing attack is an SMS message that looks like it came from your banking institution. It tells you your account has been compromised and that you need to respond immediately. The attacker asks you to verify your bank account number, SSN, etc. Once the attacker receives the information, the attacker has control of your bank account.

Vishing

Vishing has the same purpose as other types of phishing attacks. The attackers are still after your sensitive personal or corporate information. This attack is accomplished through a voice call. Hence the “v” rather than the “ph” in the name.

A common vishing attack includes a call from someone claiming to be a representative from Microsoft. This person informs you that they've detected a virus on your computer. You're then asked to provide credit card details so the attacker can install an updated version of anti-virus software on your computer. The attacker now has your credit card information and you have likely installed malware on your computer.

The malware could contain anything from a banking Trojan to a bot (short for robot). The banking Trojan watches your online activity to steal more details from you – often your bank account information, including your password.

A bot is software designed to perform whatever tasks the hacker wants it to. It is controlled by command and control (C&C) to mine for bitcoins, send spam, or launch an attack as part of a distributed denial of service (DDoS) attack.

Example of Vishing

In 2019, there was a vishing campaign that targeted members of the UK's parliament and their staffers. The attack was part of an assault that involved at least 21 million spam emails targeting UK lawmakers.

Email phishing

Email phishing is the most common type of phishing, and it has been in use since the 1990s. Hackers send these emails to any email addresses they can obtain. The email usually informs you that there has been a compromise to your account and that you need to respond immediately by clicking on a provided link. These attacks are usually easy to spot as language in the email often contains spelling and/or grammatical errors. Some emails are difficult to recognise as phishing attacks, especially when the language and grammar are more carefully crafted. Checking the email source and the link you're being directed to for suspicious language can give you clues as to whether the source is legitimate. Another phishing scam, referred to as sextortion, occurs when a hacker sends you an email that appears to have come from you. The hacker claims to have access to your email account and your computer. They claim to have your password and a recorded video of you. The hackers claim that you have been watching adult videos from your computer while the camera was on and recording. The demand is that you pay them, usually in Bitcoin, or they will release the video to family and/or colleagues.

Example of Email Phishing

Hackers used LinkedIn to grab contact information from employees at Sony and targeted them with an email phishing campaign. They got away with over 100 terabytes of data.

Search engine phishing

Search engine phishing, also known as SEO poisoning or SEO Trojans, is where hackers work to become the top hit on a search using a search engine. Clicking on their link displayed within the search engine directs you to the hacker's website. From there, threat actors can steal your information when you interact with the site and/or enter sensitive data. Hacker sites can pose as any type of website, but the prime candidates are banks, money transfer, social media, and shopping sites. Phishing Phishing Attacks Types of Phishing Smishing Social Media Phishing Related Articles Report phishing Early phishing How RSA got hacked NIST password guidance Smishing example Related Research Online phishing Identifying & Mitigating Phishing Attacks Stolen Apple credentials Sextortion.

HTTPS Phishing

An HTTPS phishing attack is carried out by sending the victim an email with a link to a fake website. The site may then be used to fool the victim into entering their private information.

Example of HTTPS Phishing

Hacker group Scarlet Widow searches for the employee emails of companies and then targets them with HTTPS phishing. When the user gets a mostly empty email, they click on the little link that is there, taking the first step into Scarlet Widow's web.

Pharming

In a pharming attack, the victim gets malicious code installed on their computer. This code then sends the victim to a fake website designed to gather their login credentials.

Example of Pharming

In 2007, a complex pharming attack went after at least 50 financial institutions across the world. Users were directed to false websites and instructed to enter sensitive information.

Pop-up Phishing

Pop-up phishing often uses a pop-up about a problem with your computer's security or some other issue to trick you into clicking. You are then directed to download a file, which ends up being malware, or to call what is supposed to be a support center.

Example of Pop-up Phishing

Users have sometimes received pop-ups saying they can qualify for AppleCare renewal, which would supposedly avail them of extended protection for their Apple devices. However, the offer is fake.

Evil Twin Phishing

In an evil twin attack, the hacker sets up a false Wi-Fi network that looks real. If someone logs in to it and enters sensitive details, the hacker captures their info.

Example of Evil Twin Phishing

A Russian military agency called GRU was recently charged with executing evil twin attacks using fake access points. The access points were made to look like they provided connections to real networks when in reality they led users to sites that stole their credentials or downloaded malware onto their computers.

Watering Hole Phishing In a watering hole phishing attack, a hacker figures out a site a group of users tends to visit. They then use it to infect the users' computers in an attempt to penetrate the network.

Example of Watering Hole Phishing

In 2012, the U.S. Council on Foreign Relations was targeted by a watering hole attack. The assault aimed to take advantage of the high-profile users that were frequenting the site, as well as the login credentials they could provide. The attack achieved some success, particularly using a vulnerability within Internet Explorer.

Clone Phishing

A clone phishing attack involves a hacker making an identical copy of a message the recipient already received. They may include something like "resending this" and put a malicious link in the email.

Example of Clone Phishing

In a recent attack, a hacker copied the information from a previous email and used the same name as a legitimate contact that had messaged the victim about a deal. The hacker pretended to be a CEO named Giles Garcia and referenced the email Mr. Garcia had previously sent. The hacker then proceeded to pretend to carry on the previous conversation with the target, as if they really were Giles Garcia.

Deceptive Phishing

Deceptive phishers use deceptive technology to pretend they are with a real company to inform the targets they are already experiencing a cyberattack. The users then click on a malicious link, infecting their computer.

Example of Deceptive Phishing

Users were sent emails that came from the address support@apple.com and had "Apple Support" in the sender information. The message claimed that the victim's Apple ID had been blocked. They were then prompted to validate their accounts by entering information the hacker would use to crack it.

Social Engineering

Social engineering attacks pressure someone into revealing sensitive information by manipulating them psychologically.

Example of Social Engineering

A hacker pretended to be a representative of Chase Bank while saying that the action was needed on the target's debit or ATM card. The attacker was trying to pressure the victim into divulging their information by leveraging their fear of not being able to access their money in their Chase account.

Angler Phishing

Anglers use fake social media posts to get people to provide login info or download malware.

Example of Angler Phishing

Hackers pretended to represent Domino's Pizza on Twitter, fielding the concerns and comments of customers. Once they engaged with a customer, they would use their situation to try to get their personal information—using the guise of trying to get them a refund or a reward.

Man-in-the-Middle (MTM) Attacks

With a man-in-the-middle attack, the hacker gets in "the middle" of two parties and tries to steal information exchanged between them, such as account credentials.

Example of Man-in-the-Middle-Attack

In 2017, Equifax, the popular credit score company, was targeted by man-in-the-middle attacks that victimized users who used the Equifax app without using HTTPS, which is a secure way to browse the internet. As the users accessed their accounts, the hackers intercepted their transmissions, stealing their login credentials.

Website Spoofing

With website spoofing, a hacker creates a fake website that looks legitimate. When you use the site to log in to an account, your info is collected by the attacker.

Example of Website Spoofing

Hackers made a fake Amazon website that looked nearly identical to the real Amazon.com but had a different Uniform Resource Locator (URL). All other details, including fonts and images, looked legitimate. Attackers were hoping that users would put in their username and password.

Domain Spoofing

Domain spoofing, also referred to as DNS spoofing, is when a hacker imitates the domain of a company—either using email or a fake website—to lure people into entering sensitive information. To prevent domain spoofing, you should double-check the source of every link and email.

Example of Domain Spoofing

An attacker would execute a domain spoofing attack by creating a fraudulent domain made to look like a real LinkedIn site, for example. When users go to the site and enter any information, it is sent straight to hackers who could use it or sell it to someone else.

Image Phishing

Image phishing uses images with malicious files in them meant to help a hacker steal your account info or infect your computer.

Example of Image Phishing

Hackers have made use of AdGholas to hide malicious code written in JavaScript inside images and HTML files. When someone clicked on an image generated by AdGholas, malware would be downloaded onto their computer that could be used to phish for their personal information.

Search Engine Phishing

A search engine phishing attack involves an attacker making fake products that look attractive. When these pop up in a search engine, the target is asked to enter sensitive information before purchasing, which then goes to a hacker.

Example of Search Engine Phishing

In 2020, Google said that they found 25 billion spam pages every day, like the one put up by hackers pretending to be from the travel company Booking.com. An ad would pop up in users' search results that looked like it was from booking.com and included the site's address and the kind of wording users would expect from a real ad by the company. After users clicked, they were prompted to enter sensitive login information that was then transmitted to hackers.

Phishing attacks have become a common phenomenon since the inception of the internet back in the '90s. Although they intrude on the personal information of the victims, the right knowledge and preparation can act as robust phishing protection measures. Follow these guidelines to learn on how to avoid phishing:

Keeping Updated With The Latest Phishing Techniques

Hackers continuously invent new techniques, and they also keep updating the existing ones to trick more targets. Without the knowledge of these continually updating phishing techniques, a user can easily fall prey to one. Enterprises need to ensure thorough awareness drives, deploy the right countermeasures, and train the employees on their crucial role in information security.

Thinking Twice Before Clicking

Clicking on the links in random and suspicious emails can prove to be costly. A phishing email typically claims to be from a legitimate enterprise and contains a link that leads to a site which looks exactly like the original one. If an unsuspecting user enters his/her details on the website, the hackers gain access to these private credentials. Thus, one must think twice before clicking on such links. One simple safeguard, though not foolproof, is to hover over the links before clicking them. The destination website displayed can usually help decide whether the site is authentic or fake. Also, malicious emails never address the users by their names. This is because the attackers are not yet in possession of such details, and the email is most likely one of the thousands sent to other people. Thus, if one receives an email that starts with generic greetings like “Dear customer,” it should serve as a red flag, and they must be vigilant.

Installing An Anti-Phishing Toolbar

It is a popular measure that users deploy to prevent phishing. Most popular internet browsers provide the option for anti-phishing toolbars. These toolbars run routine checks on the visited websites and compare them with the known phishing sites in their database. If a user, accidentally or otherwise, navigates to a malicious website, the toolbar alerts them.

Verifying A Site’s Security

When supplying sensitive information to the website, it is but natural to be a little wary. The vital checks for a secure website are:
Ensuring that the site’s URL begins with https.
Looking for a closed lock icon near the address bar. Checking the site’s security certificate.
It is prudent to not download any files or attachments from suspicious websites. Many times, even search engines throw up links to a phishing website.

Checking Online Accounts Regularly

As an anti-phishing security measure, one must check in with each of their online accounts regularly. Regularly changing the passwords to online accounts is an effective phishing protection measure. Doing so will prevent many attacks, including bank and credit card phishing scams. Additionally, regularly checking bank account statements is also a sensible measure. To ensure that there are no fraudulent transactions from their account, users must check every entry in their monthly financial statements.

Keeping The Browser Up To Date

Hackers discover and exploit security loopholes in operating systems and browsers to carry out hyper-targeted phishing attacks. This is one of the reasons that popular browsers release security patches from time to time. One must download and install the security update, advisably as soon as it is available.

Using Firewalls

Internet users must deploy firewalls to keep their systems inaccessible for phishers and attackers. There are two essential firewalls – a desktop firewall and a network firewall. While the former is a software, the latter is a hardware anti phishing solution. Even though most users use one or the other, it is advisable that you use them together. These firewalls act as buffers between the user, computer, and the outside intruders.

Never Giving Out Personal Information

One must avoid sharing personal or financially sensitive information over the internet. Whenever in doubt, make it a habit to visit the business's website, note down their contact details, and give them a call. Most phishing emails redirect users to phishing websites that ask them for financial or personal information. Never share confidential data through the links in emails. If you have to do so, open the official website of the alleged organization by typing the address in your browser, and enter the required details there only. In addition to this precaution, remember that hackers can intercept and misuse any sensitive information which is present in emails. Hence, one should refrain from sending emails containing personal information.

Get free anti-phishing add-ons

Most browsers nowadays will enable you to download add-ons that spot the signs of a malicious website or alert you about known phishing sites. They are usually completely free so there's no reason not to have this installed on every device in your organization.

Rotate passwords regularly

If you've got online accounts, you should get into the habit of regularly rotating your passwords so that you prevent an attacker from gaining unlimited access. Your accounts may have been compromised without you knowing, so adding that extra layer of protection through password rotation can prevent ongoing attacks and lock out potential attackers.

Don't ignore those updates

Receiving numerous update messages can be frustrating, and it can be tempting to put them off or ignore them altogether. Don't do this. Security patches and updates are released for a reason, most commonly to keep up to date with modern cyber-attack methods by patching holes in security. If you don't update your browser, you could be at risk of phishing attacks through known vulnerabilities that could have been easily avoided.

Effect of phishing attacks:

Phishing attacks are gaining momentum because they are easy to set up, rewarding, and pose little risk to cybercriminals. It can be as simple as hosting a fake webpage or malicious file and sending spoofed emails to victims and waiting for stolen access or data. Cybercriminals employ two approaches to phishing. The more common approach is general phishing which involves mass fake email campaigns in the hope of getting as many victims as possible. The other method is spear phishing where attackers customize phishing emails to their target in order to increase the chance of success. The cost of a phishing attack can be grave depending on the attack scope. We discuss some of the ways that phishing attacks affect businesses below:

1. Loss of Data

Clicking on a malicious link in an email can hand over the data and system of an organization to a hacker. They are then free to do what they want including theft for further criminal purposes, corruption, and deletion. Data loss is considered the most severe effect of phishing attacks.

2. Damaged Reputation

Companies suffer reputation loss following a data breach executed through phishing attacks. Announcement of a breach leads to loss of trust for the company among the general public. Regardless of an organization's previous standing, data breaches exert a strong negative effect on its brand and it may be seen as untrustworthy for a long time following a successful hack. It could induce public backlash against a company for not doing enough to protect user's data.

3. Direct Monetary Loss

Extra funds will be needed to manage identity protection, compensation of customers or employees whose data was stolen following a phishing attack. Funds could also be transferred out from a company's account through impersonation via phishing.

4. Loss of Productivity

Data breaches or system compromise arising from phishing attacks cause business disruption. Following a successful phishing attack, a large part of a business' time will be spent on trying to recover lost data and investigating the breach with little left for actual business. Employees' productivity will also take a hit as many systems are put offline for reconfiguration and cleaning.

5. Loss of Customers

Successful phishing attack scares customers away from a business. A UK survey revealed that more than half of consumers stop patronizing a hacked organization for several months after a data breach. Some 41% of customers no longer patronize businesses that got their data leaked. This effect could haunt an organization for a long time.

6. Financial Penalties

When sensitive customers' data end up in the public domain, the affected business is held responsible. In addition to the direct monetary loss from failure to defend against phishing, heavy regulatory fines can be placed on an organization for mishandling customer's data. The penalties target businesses that don't follow best practices for protecting their customer's private data. Violating regulatory requirements such as HIPAA, PCI, and European GDPR may attract heavy fines. The extent of the fines depends on the industry and the scope of the breach.

7. Intellectual Property Theft

A business asset isn't just money or equipment, intellectual property could even be more important. Intellectual property may be stolen through phishing attacks and could even be the motivation for the attack in the first place. Heavy investment goes into research and development, new technology as well as trade secrets. When these are compromised, they could setback the business involved and make them less competitive.

8. Loss of Company Value

Phishers can also cost a company a significant part of its market value as a result of the loss of investors' confidence. Some investors would no longer trust the affected organization and may move their funds elsewhere to protect their portfolio. A successful phishing attack can have multiple negative effects on an organization. This may include data loss, compromised credentials, ransomware, and malware infestation. It is pertinent that you prioritize employee cybersecurity education, install advanced security solutions and implement policies that will block phishing attempts and protect your business from its impacts.

1.4 Objectives.

- To collect URL data for phishing and legitimate websites.
- Preprocess and clean the data for ML model training.
- Feature extraction from the URL data.
- Train ML models using the dataset prepared.
- Train KNN, Decision Tree, Random Forest..
- Test the models using the evaluation metrics and tabulate their performance.
- Deploy the model at the backend of a chrome extension.
- Do the real-time testing of the chrome extension and conclude.

1.5 Motivation.

- Phishing is one of the most common threats in daily cyber life.
- URL phishing is an addressable problem with advanced and doable technologies like AI/ML.
- Simple models developed using ML and DL techniques can be used to detect phishing accurately.

Chapter 2

LITERATURE SURVEY

The literature survey has been conducted on different papers such as technical papers and review papers in the domain published in leading publications, journals and conferences. Keywords such as phishing, phish detection, machine learning, deep learning, etc are used to filter out.

This chapter discusses the previous and existing works on malicious URL detection with different approaches. Jain, A.K. and Gupta, B.B., observed that attackers steal sensitive information such as personal identification number (PIN), credit card details, log in, password, etc., from Internet users. In this paper, the author has proposed a machine-based reading program based on the Uniform Resource Locator (URL) features. To test the performance of the proposed system, the author has taken 14 features in the URL to find the website as sensitive identity theft or non-sensitive identity theft crime [11, 1]. The proposed approach is being trained using sensitive identity theft and official URLs with SVM and Naïve Bayes divisions. Test results show 90% accuracy in detecting identity theft websites using the SVM separator [1].

Purbay M., and Kumar D. examined multiple machine learning methods for obtaining URLs by analyzing various URL parts using machine learning and in- depth learning methods. The authors discussed different ways of reading surveillance to identify criminal URLs that steal sensitive information based on dictionary, WHOIS architecture, PageRank, traffic level information, and key page layouts. Learn how the volume of different training data influences the accuracy of class dividers. The research includes Vector Support Machine (SVM), K-NN, random forest classification (RFC), and Artificial Neural Network (ANN) classification methods [19, 3]. Gandotra E., and Gupta D conducted a comparative study on machine learning based on the outputs and operational selections. They studied 6157 incorrect pages and found several Machine learning methods have been used for best results. The latter job selection method is used to maximize model performance [3, 12]. The random forest algorithm gained accuracy before and after selecting features and significantly increased construction time. Experimental results have shown that using a selective method of machine learning algorithms can improve the performance of classification models to detect the crime of stealing sensitive information without reducing its effectiveness [2].

M. Abutaha et al. developed URLNet, a Convolutional Neural-Network (CNN) based on an in-depth reading framework that uses alphabetical characters and URLs to capture semantic information to distinguish malicious and dangerous URLs. Their work has demonstrated a promising approach to URL acquisition through in-depth reading. They discussed the limitations of features obtained using the word bag and mathematical features such as the length of the different segments in the URL. Use CNN to get useful structural information for URLs with two separate databases generated by letters and URL names. Word Level CNN is similar to CNN characters level except that convolution operators are used in words [17, 15]. Database URLs are collected from VirusTotal [3]. They have created a feature set using a training corporation with all the unique words as a dictionary. This method provides another way to separate malicious URLs by capturing a few semantic information via URLNet, which are existing methods based on word tag elements that could not. It provides an essential escape from the AUC beyond the foundation [3].

S. Alrefaai J. et al., The author investigated how the URLs of identity theft can be categorized in a set of URLs containing incorrect URLs. They discuss random signal engineering, feature extraction using host-based analysis, and mathematical analysis. In a comparative study, several class dividers were used and found that the results for all the different class dividers were almost identical. Authors argue by suggesting an easy way to remove functionality from URLs with simple common words. Other factors can be tested that lead to better results. The database used in the study includes old URLs. Thus, there is a possibility of inefficiency [4]. B. Geyik et al. introduce the CBR-PDS. It relies heavily on the CBR method as the core component. The system is flexible and flexible as it can quickly adapt to detecting cybercrime attacks with a small amount of data set compared to other detectors requiring extensive training in advance [18]. Authors test their system using different scenarios for 572 phishing and official URLs. Studies show that the accuracy of the CBR-PDS system exceeds 95.62%, yet it significantly improves the accuracy of categories with a small set of features and limited data sets [5].

S. Singh et al. have proposed a clever way to detect sensitive identity theft in 2015 called PhishShield, a desktop application that focuses on detecting identity theft using URLs and website content of sensitive identity theft websites [6]. The features released by PhishShield are minimal link links, zero links in the HTML body, copyrighted content, title content, and website logos. PhishShield is faster, more accurate, and has a broader range of access to criminal websites to steal sensitive information compared to the blacklisting and whitelisting system [7, 14]. However, detection effectiveness decreases when the attacker understands the heuristic process and can successfully pass the heuristic filter. they demonstrated a solution to detect identity theft using Convolutional Neural Network (CNN) character-based analysis of website URLs using a model based on fast-paced learning solutions. Their model does not include using services from third parties or retrieving content from the targeted website. They capture sequence patterns and URL unit information without the need for an idea about the crime of stealing sensitive information in advance. Consecutive patterns quickly classify the original URL. They also compare different traditional and in-depth machine learning models. Feature sets include handicrafts, embedded characters, Term's Frequency-Inverse Document Frequency (TF-IDF), and calculation vector features at the character level [20]. The experimental results of Aljofey et al. have brought 95.02% accuracy to their database from the proposed model. Benchmark databases work better than current sensitive identity theft URLs models that produce 98.58% accuracy, 95.46%, and 95.22% accuracy [6]. J. Kumar et al. used a productive argument network to classify URLs into categories and bypass criminals to steal sensitive information based on restricted lists. In addition, the researchers argued that the system could surpass both simple ML acquisition strategies and novice ones [7]. S. Parekh et al. published “Phishing Website Classification and Detection Using Machine Learning”. By making use of lexical structure URL to classify url into different parts and identify the Url whether the given url is phishing url or not. In, this paper, they have compared different machine learning techniques for the phishing URL classification task and achieved the highest accuracy of 96% for Naïve Bayes Classifier with a precision=1, recall = .95 and F1-Score= .96. There are many techniques to overcome tricked by phishing website. One of the methods mostly used to detect phishing is by using visual similarity. This method is to dissimilar phishing webpage, which also reduce the successful rate of victim got tricked by phishing scams. Besides that, there is another method to detect phishing website is by using compression algorithm. Compression algorithm is a critical component which perform a compression of nine compressors that include 1-dimensional string and 2-dimensional image compression [18]. The main aim of this paper is to spot phishing attacks that connects the victim's email by mistreatment by applying decision tree algorithm that enforced within the application. This project mainly focused on detect on attachment file of phishing website in the email buddies by using decision tree algorithm. Anti-Phishing detection application used to detect, identify and block the phishing website or email that effected by the phishing website. It is able to calculate the percentages of stored phishing emails in the user's email [8]. H. Yuan et al. published "Detecting Phishing Websites Using Machine Learning". The system acts as an extra functionality to a web browser as an extension that mechanically notifies the user once it detects a phishing website. The system is predicated on a machine learning method, notably supervised learning. They've selected the Random Forest technique because of its sensible performance in classification. The focus will be on the features combination that we get from Random Forest (RF) technique, as it has good accuracy, is relatively robust, and has a good performance. Recently, there have been several studies that are trying to solve the phishing problem. They can be classified into four types: blacklist, heuristic, content analysis, and machine learning techniques. The blacklisting technique compares the URL with an existing database that contains a list of phishing website URLs. Because of the rapid increase of such phishing attacks, the blacklist approach has become more inefficient in checking whether each URL is a phishing website or not, and this kind of delay can also lead to zero-day attacks from these new phishing sites [9]. Rishikesh Mahajan et al. compared phishing mitigation techniques, such as blacklist, heuristics, visual similarity, and machine learning and concluded that these techniques have limitations in dealing with zero-hour attacks and proactive detection of phishing websites. The authors proposed suspicious URL's generation and to predict likely phishing sites from the given legitimate brand domain name and scores and judge suspects by calculating various indexes to detect phishing websites [18].

Chapter 3

EXISTING SYSTEM

There have been many technologies applied to combat phishing attacks ever since its inception. Phishing, a prevalent cyber threat, continues to evolve in sophistication and frequency, necessitating the development and deployment of robust detection systems. Several existing systems employ diverse techniques and methodologies to combat phishing attacks effectively. This section provides an in-depth exploration of various existing systems categorized based on their detection mechanisms.

3.1 Black-Listing Systems

Black-listing systems constitute a pivotal component of phishing detection mechanisms, leveraging a proactive approach to thwart malicious activities. Operating on the principle of precluding access from known malicious sources, these systems maintain extensive databases comprising identified phishing entities such as fraudulent websites, email addresses, IP addresses, and URLs. In essence, black-listing serves as a defensive barrier, fortifying organizational networks and systems against infiltration attempts from recognized threats. One of the fundamental pillars of black-listing systems is their reliance on threat intelligence feeds, which provide real-time updates on emerging phishing campaigns, compromised domains, and malicious IP addresses. These feeds serve as the cornerstone of black-listing databases, furnishing organizations with invaluable insights into the evolving threat landscape and enabling proactive mitigation of potential risks. Through continuous monitoring and analysis of threat intelligence feeds, black-listing systems remain adept at identifying and cataloging new phishing entities, ensuring their databases remain up-to-date and comprehensive. Central to the functionality of black-listing systems is the process of automated verification and validation of incoming network traffic against the black-listed entities. Upon receipt of network requests or communication attempts, these systems swiftly cross-reference the source entities with their black-listing databases to ascertain their legitimacy. Any matches or correspondences with known malicious entities trigger immediate blocking actions or alerts, thwarting potential phishing attempts and safeguarding organizational assets from compromise. Moreover, black-listing systems often incorporate dynamic and configurable blocking policies, enabling organizations to tailor their response strategies based on specific threat profiles, risk tolerance levels, and operational requirements. Administrators can define granular rules and thresholds for black-listing actions, such as blocking access to specific URLs, quarantining suspicious emails, or redirecting traffic through secure gateways for further inspection. This flexibility empowers organizations to enforce stringent security measures while minimizing disruptions to legitimate business operations. Despite their efficacy in mitigating known phishing threats, black-listing systems confront certain inherent limitations and challenges. Chief among these is the inability to preemptively detect and block zero-day or previously unseen phishing attacks, which may exploit undiscovered vulnerabilities or evade conventional black-listing measures. Additionally, the reliance on threat intelligence feeds poses a risk of false positives or inaccurate classifications, potentially leading to the inadvertent blocking of benign entities or legitimate communications.

3.2 Heuristic-Based Systems

White-listing systems, a fundamental component of cybersecurity infrastructure, function as a proactive defense mechanism against phishing attacks by selectively permitting access only to trusted entities. Operating on the principle of trust verification, these systems maintain extensive databases comprising authenticated and vetted sources, including known legitimate websites, email addresses, domains, IP addresses, and applications. By restricting access exclusively to entities listed within the white-listing database, organizations bolster their defenses against unauthorized access attempts and mitigate the risk of falling victim to phishing scams. At the core of white-listing systems lies the process of identity verification and validation, wherein incoming network traffic or communication attempts are scrutinized and authenticated against the entries within the white-listing database. Upon receipt of network requests or access attempts, these systems employ sophisticated algorithms and verification mechanisms to ascertain the legitimacy of the source entities. Any matches or correspondences with pre-approved entries within the white-listing database signal trustworthiness and prompt authorization of access, facilitating seamless communication and interaction while minimizing exposure to potential threats. One of the key advantages of white-listing systems is their ability to provide granular control and customization over access permissions and authorization policies. Administrators can define and configure specific criteria for inclusion within the white-listing database, tailoring the system's behavior to align with organizational security requirements, risk tolerance levels, and operational needs. This flexibility empowers organizations to enforce stringent access controls while accommodating legitimate business processes and workflows, thereby striking a balance between security and productivity. Furthermore, white-listing systems often integrate additional authentication and validation mechanisms to enhance the reliability and integrity of the verification process. Multi-factor authentication (MFA), digital signatures, cryptographic certificates, and other identity verification techniques may be employed to augment the trustworthiness of approved entities and mitigate the risk of impersonation or spoofing attempts. By incorporating multiple layers of authentication, white-listing systems fortify their defenses against sophisticated phishing tactics and unauthorized access attempts. Despite their efficacy in mitigating known phishing threats, white-listing systems encounter certain challenges and limitations. Chief among these is the inherent difficulty in maintaining an exhaustive and up-to-date white-listing database, particularly in dynamic and rapidly evolving environments where entities may change status or ownership frequently. Additionally, the reliance on pre-approved entries within the white-listing database may inadvertently exclude legitimate entities or sources that have not been vetted or authenticated, potentially leading to access restrictions or disruptions in communication.

3.3 Hybrid Approaches

Hybrid approaches to phishing detection represent a sophisticated integration of multiple detection mechanisms and strategies, synergistically combining the strengths of diverse methodologies to enhance detection accuracy and resilience against evolving threats. These approaches leverage a combination of rule-based heuristics, machine learning algorithms, behavioral analysis techniques, and threat intelligence feeds to create a comprehensive defense framework capable of effectively identifying and mitigating phishing attacks. At the core of hybrid approaches lies the concept of synergy, wherein each detection mechanism complements and reinforces the capabilities of the others to form a cohesive and adaptive defense strategy. Rule-based heuristics serve as an initial line of defense, leveraging predefined rules and patterns to flag suspicious activities or entities indicative of phishing attempts. These rules may include criteria such as suspicious email content, malformed URLs, mismatched sender information, or unusual network behaviors. While rule-based heuristics offer speed and simplicity in detection, they may lack the adaptability to identify novel or sophisticated phishing tactics. To augment the capabilities of rule-based heuristics, hybrid approaches incorporate machine learning algorithms that leverage historical data and pattern recognition techniques to detect subtle, nuanced indicators of phishing attacks. Supervised learning algorithms, such as support vector machines (SVMs) or random forests, analyze labeled datasets to identify patterns and features associated with phishing attempts, enabling them to generalize and detect previously unseen threats with high accuracy. Unsupervised learning techniques, such as clustering or anomaly detection, further enhance detection by identifying deviations from normal behavior patterns indicative of phishing activity. Behavioral analysis techniques play a crucial role in hybrid approaches, enabling the detection of anomalous behaviors or deviations from established norms that may signify phishing attempts. These techniques monitor user interactions with emails, websites, or applications to identify suspicious activities, such as clicking on malicious links, entering sensitive information on phishing websites, or exhibiting erratic browsing behaviors. By analyzing behavioral patterns and anomalies, hybrid approaches can detect and mitigate phishing attacks in real-time, minimizing the risk of data breaches or security incidents.

3.4 Behavioral Analysis

Behavioral analysis stands as a pivotal component in the arsenal of phishing detection strategies, employing a proactive approach to identify and mitigate threats by scrutinizing user behaviors and interactions with digital assets such as emails, websites, or applications. Unlike traditional detection methods that rely solely on static indicators or patterns, behavioral analysis techniques delve into the dynamic aspects of user actions and behaviors, enabling the detection of subtle anomalies or deviations indicative of phishing attempts.

At the heart of behavioral analysis lies the premise of establishing baseline behavior profiles for individual users or entities, which serve as reference points for identifying deviations or aberrations. These baseline profiles are constructed through continuous monitoring and analysis of user interactions over time, capturing normal patterns of behavior across various dimensions such as email engagement, web browsing activities, file access behaviors, and application usage. By establishing a comprehensive understanding of typical user behaviors, behavioral analysis techniques can effectively differentiate between legitimate activities and suspicious actions that may signify phishing attempts.

One of the key advantages of behavioral analysis is its adaptability and resilience against evolving phishing tactics and strategies. Unlike static rule-based approaches that may struggle to keep pace with rapidly changing attack vectors, behavioral analysis techniques have the flexibility to adapt and evolve alongside emerging threats. By continuously learning and updating behavior profiles based on real-time observations, these techniques can detect new and previously unseen phishing tactics, including sophisticated social engineering schemes or targeted spear-phishing campaigns.

Behavioral analysis encompasses a diverse array of techniques and methodologies, ranging from simple anomaly detection algorithms to advanced machine learning models. Anomaly detection algorithms analyze user behavior patterns and identify deviations or outliers that deviate significantly from established norms. These anomalies may manifest as unusual patterns of email communication, irregular web browsing activities, or atypical file access behaviors, warranting further investigation or intervention. Machine learning models, such as neural networks or ensemble classifiers, leverage labeled datasets to identify subtle patterns and correlations within behavioral data, enabling more accurate detection of phishing attempts based on nuanced features and indicators.

3.5 URL Analysis

URL analysis is a crucial component of phishing detection strategies, providing organizations with a proactive mechanism to assess the legitimacy and trustworthiness of web addresses and links encountered in emails, messages, or websites. By scrutinizing the characteristics and attributes of URLs, URL analysis techniques aim to identify indicators of phishing attempts, malicious activities, or fraudulent websites, thereby enabling users to make informed decisions and avoid potential security risks. At the core of URL analysis lies the process of assessing various aspects of URLs, including their structure, syntax, domain registration details, hosting information, and presence of suspicious or malicious elements. One of the primary indicators of phishing URLs is their deceptive or misleading nature, often designed to mimic legitimate websites or services in an attempt to deceive users. URL analysis techniques leverage pattern recognition algorithms, lexical analysis, and heuristic rules to identify common characteristics of phishing URLs, such as misspelled domain names, subdomain anomalies, excessive hyphens or numerical characters, and unusual top-level domains (TLDs). Furthermore, URL analysis techniques often incorporate reputation-based assessments to evaluate the trustworthiness and reputation of web domains and URLs. Reputation-based analysis relies on databases of known malicious domains, blacklists, or threat intelligence feeds to identify URLs associated with phishing attacks, malware distribution, or fraudulent activities. By cross-referencing URLs against these databases, URL analysis systems can flag suspicious or malicious links and warn users of potential security risks before accessing the corresponding websites or resources. Moreover, URL analysis encompasses the detection of phishing URLs based on their hosting infrastructure and characteristics. Phishing websites are often hosted on compromised or malicious servers, characterized by suspicious IP addresses, hosting providers, or geographical locations. URL analysis techniques leverage geolocation data, WHOIS information, and IP reputation services to assess the hosting infrastructure of URLs and identify indicators of malicious intent. By analyzing hosting attributes and identifying anomalies or discrepancies, URL analysis systems can detect and block access to phishing websites, mitigating the risk of data breaches or credential theft.

3.6 Email Header Inspection

Email header inspection is a critical aspect of phishing detection and cybersecurity, offering insight into the origin, routing, and authenticity of email messages. By analyzing the metadata embedded within email headers, organizations can uncover valuable information about the sender, message path, and potential indicators of phishing attempts or fraudulent activities. Email header inspection techniques play a pivotal role in phishing detection strategies, enabling organizations to identify suspicious or malicious emails and protect users from falling victim to phishing scams. At the core of email header inspection lies the examination of various fields and attributes contained within the email header, including the From, To, Subject, Date, and Message-ID fields, among others. These fields provide essential information about the sender, recipient, message content, and transmission details, offering valuable insights into the legitimacy and trustworthiness of the email message. By scrutinizing these fields, email header inspection techniques can identify anomalies, inconsistencies, or suspicious patterns that may signify phishing attempts or unauthorized activities. One of the primary indicators of phishing emails is the presence of spoofed or forged sender information within the email header. Phishers often manipulate the From and Reply-To fields to impersonate trusted entities or organizations, deceiving recipients into believing that the email originated from a legitimate source. Email header inspection techniques leverage domain authentication protocols such as SPF (Sender Policy Framework), DKIM (DomainKeys Identified Mail), and DMARC (Domain-based Message Authentication, Reporting, and Conformance) to verify the authenticity of sender domains and detect email spoofing attempts. By validating the cryptographic signatures and domain alignment within email headers, organizations can ascertain the legitimacy of sender identities and mitigate the risk of phishing attacks.

3.7 Content-Based Filtering

Content-based filtering is a fundamental technique in phishing detection, enabling organizations to analyze the content of email messages and identify indicators of phishing attempts or fraudulent activities. By scrutinizing the textual and visual elements within email messages, content-based filtering techniques aim to detect suspicious patterns, malicious URLs, phishing links, or deceptive content that may signify phishing attempts. Content-based filtering plays a crucial role in phishing detection strategies, providing organizations with a proactive mechanism to assess the legitimacy and trustworthiness of email messages and protect users from falling victim to phishing scams. At the core of content-based filtering lies the analysis of various components and attributes within email messages, including the subject line, body text, attachments, embedded links, and multimedia content. These elements offer valuable insights into the intent, context, and characteristics of email messages, enabling content-based filtering techniques to identify anomalies, inconsistencies, or suspicious patterns indicative of phishing attempts. By analyzing the semantic, syntactic, and structural aspects of email content, content-based filtering systems can detect phishing emails and mitigate the risk of security breaches or data exfiltration. One of the primary indicators of phishing emails is the presence of deceptive or misleading content designed to deceive recipients into disclosing sensitive information or performing unauthorized actions. Content-based filtering techniques leverage natural language processing (NLP) algorithms, text analysis tools, and linguistic models to analyze the textual content of email messages and identify suspicious patterns or linguistic cues associated with phishing attempts. These techniques can detect phishing emails containing phishing lures, urgent requests for information or action, grammatical errors, spelling mistakes, or inconsistent language usage, which are commonly employed by phishers to manipulate and deceive recipients.

3.8 Domain Reputation Services

Domain reputation services play a pivotal role in phishing detection and cybersecurity, providing organizations with valuable insights into the trustworthiness, reputation, and legitimacy of domain names associated with email messages, websites, or online resources. By analyzing domain reputation data and assessing the historical behavior and characteristics of domains, these services enable organizations to identify indicators of phishing attempts, malicious activities, or fraudulent behavior, thereby mitigating the risk of security breaches and protecting users from falling victim to phishing scams. At the core of domain reputation services lies the analysis of various attributes and characteristics associated with domain names, including registration details, hosting infrastructure, historical usage patterns, and presence on blacklists or threat intelligence feeds. These services leverage domain intelligence databases, threat feeds, and reputation scoring algorithms to evaluate the reputation and trustworthiness of domains and assess their potential risk levels. By cross-referencing domain attributes against known indicators of malicious behavior, domain reputation services can flag suspicious or high-risk domains and warn users and administrators of potential security risks. One of the primary functions of domain reputation services is to assess the registration and ownership details of domains to identify potential signs of malicious intent or fraudulent activity. Phishers often register domains with deceptive or misleading names, intending to impersonate legitimate organizations or entities and deceive users into disclosing sensitive information or performing unauthorized actions. Domain reputation services analyze domain registration data, WHOIS information, and historical ownership records to identify anomalies, inconsistencies, or red flags indicative of domain abuse or illicit activities. By scrutinizing registration details and identifying suspicious patterns, these services can flag potentially malicious domains and mitigate the risk of phishing attacks.

3.9 Sandbox Analysis

Sandbox analysis is a critical component of cybersecurity and malware detection, providing organizations with an effective means of analyzing and evaluating the behavior of suspicious files, applications, or code in a controlled environment. By executing potentially malicious software within a secure and isolated sandbox environment, organizations can observe its actions, interactions, and effects on system resources without risking the integrity or security of their network infrastructure. Sandbox analysis techniques play a pivotal role in identifying and mitigating the threat posed by malware, zero-day exploits, and advanced persistent threats (APTs), enabling organizations to proactively defend against cyberattacks and protect sensitive information and assets. At the core of sandbox analysis lies the creation and deployment of secure, isolated environments known as sandboxes, where suspicious files or executables can be executed and monitored in a controlled manner. Sandboxes typically consist of virtualized or containerized environments that emulate the underlying operating system and runtime environment, allowing malware samples to run in an environment that closely resembles the target system while isolating them from the host environment. By isolating potentially malicious code within a sandbox environment, organizations can observe its behavior, analyze its actions, and assess its impact on system resources without exposing their network infrastructure to the risk of infection or compromise. One of the primary objectives of sandbox analysis is to observe and monitor the behavior of suspicious files or executables as they execute within the sandbox environment. Sandboxes capture detailed information about the runtime behavior of malware samples, including file system modifications, registry changes, network communications, process creation, and system calls. By monitoring these activities in real-time, organizations can identify malicious behaviors, anomalous activities, or indicators of compromise associated with malware infections. Sandbox analysis techniques leverage advanced monitoring and instrumentation capabilities to capture and record relevant data points, enabling analysts to gain insights into the behavior and functionality of malware samples.

Chapter 4

PROPOSED SYSTEM

The problem here is the lack of a light weight system for detecting phishing websites. Website phishing is very common and phishing attackers are adopting advanced technologies like AI/ML to escape from defense mechanisms. So it's a must to develop a phishing detection system that uses these technologies for detection and that is lightweight. The system or the model should also be accurate enough. The developed system should input the URL as a plain text and should output whether the URL is phishing or legitimate in no time.

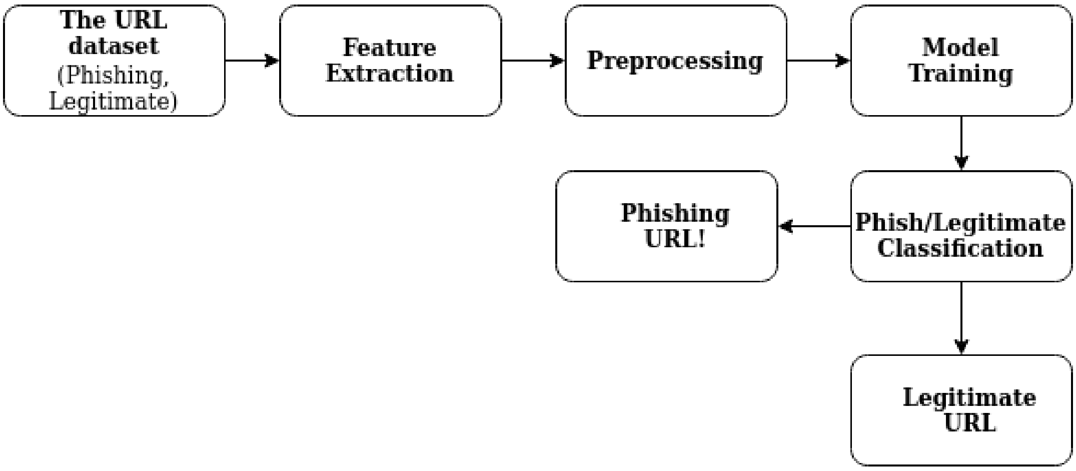


Figure 4.1: Data Flow Diagram of the System Design

The data flow diagram in the figure shows the process of phishing URL detection in brief. The URL dataset contains URLs in the two classes: phishing and legitimate. This is fed to the preprocessing and feature extraction then to the model training process. The figure 3 shows a detailed explanation of the process.

The proposed methodology has steps like below:

1. Phishing Detection : The dataset is prepared using web scraping various websites such as www.kaggle.com and www.phishtank.com. Then it is preprocessed and features are extracted to make the final dataset. This dataset is used to train teh ML model and the final model is saved as pickle file for future use.

The problem here is a binary classification machine learning problem. The machine learning model has to input the URL and predict whether it is a phishing URL or a legitimate URL. There are many ways to do this such as detection from text content, from website components, etc. But URL-based detection proved to be more efficient, effective and accurate. To train a ML model for detecting Legitimate and phishing URLs from a set of unknown URLs, we have to collect a number of such URLs in both phishing and Legitimate categories.

Framing an ML problem:
Input : Website URL ; Output : Phishing OR Legitimate.
Input → text-related features of a URL (eg: length of URL, presence of @ symbol, etc) →ML model → Prediction

Machine Learning task: Supervised Learning.
Binary classification task - (Two classes to predict, legitimate and phishing)

Steps involved:

1. To collect URL data for phishing and legitimate websites.
2. Preprocess and clean the data for ML model training.
3. List out the features to be extracted.
4. Feature extraction from the URL data.
5. Train ML models using the dataset prepared.
6. Develop Decision Tree, Random Forest, and XGBoost.
7. Train the ML models.
8. Test the models using the evaluation metrics and tabulate their performance.
9. Develop and deploy the Chrome extension for phishing detection.
10. Test the extension in real-time.

Chapter 5

SYSTEM DESIGN AND ARCHITECTURE

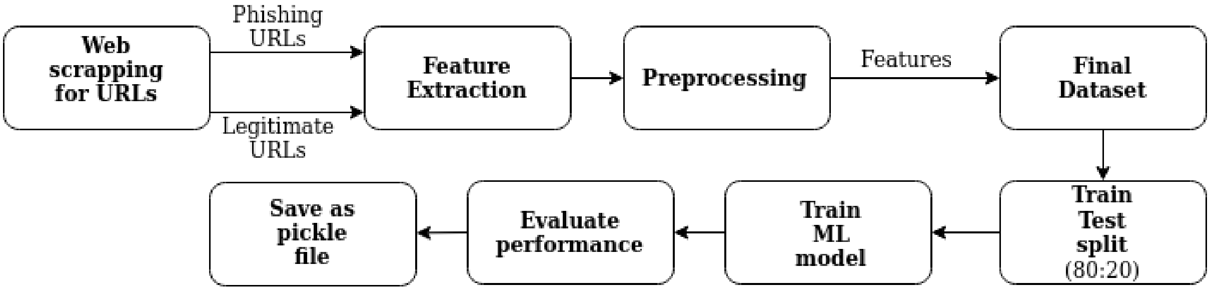


Figure 5.1 : Training and Testing

The System architecture shows the whole process that starts with the data collection of both legitimate and genuine URLs and feature extraction, preprocessing, ML development, training, evaluation, and testing.

5.1 FEATURE EXTRACTION

Features from the URLs can be extracted in many ways. We have done three types of feature extraction.

1. Address bar-based features.
2. Domain-based features.
3. HTML and Javascript-based features.

Address bar-based features:

Address bar features are related to the URL address like presence of '@' in the URL

We have to extract 9 address related features as below:

1. Domain of URL
2. IP Address in URL
3. "@" Symbol in URL

4. Length of URL
5. Depth of URL
6. Redirection "//" in URL
7. "http/https" in Domain name
8. Using URL Shortening Services “TinyURL”
9. Prefix or Suffix "-" in Domain

Details of the features:

1. Domain of URL

Domain in a URL is anything after “http://www.”

Eg: domain of “http://www.facebook.com” is “facebook.com”.

2. IP Address in URL

Check whether there is any IP address present in the URL (such as 180.256.2.0) Attackers usually use IP address in the URL instead of domain. If the domain part of the URL has an IP address, the value assigned to this feature is 1 or else 0.

3. "@" Symbol in URL

@ Symbol is a frequent presence in the Phishy URLs If the URL has '@' symbol, the value assigned to this feature is 1 or else

4. Length of URL

Phishing attacks usually use long URLs to hide the doubtful part in the address bar. So we are fixing a threshold length of 54 If the length of URL >= 54 , the value assigned to this feature is 1 or else 0.

5. Depth of URL

This feature calculates the number of sub pages in the given url based on presence of '/'.

6. Redirection "//" in URL

Presence of '//' in the URL means the user will be redirected to somewhere else But // usually comes at 6 th position in HTTP and 7th position in HTTPS So we find the presence of // in positions after 6th or 7th If the "//" is anywhere in the URL apart from after the protocol, the value assigned to this feature is 1 or else 0.

7. "http/https" in Domain name

The phishers may add the “HTTPS” token to the domain part of a URL in order to trick users. If the URL has "http/https" in the domain part, the value assigned to this feature is 1 or else 0.

8. Using URL Shortening Services “TinyURL”

URL shortening is a method in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an “HTTP Redirect” on a domain name that is short, which links to the webpage that has a long URL. If the URL is using Shortening Services, the value assigned to this feature is 1 or else 0.

9. Prefix or Suffix "-" in Domain

If the URL has '-' symbol in the domain part of the URL, the value assigned to this feature is 1 or else 0.

The next is to extract the domain-based features:

Domain-based features are related to the domain, the date of purchase and the date of expiration of the domain, owner of the domain, etc. This is done with the help of the python-whois library which is a library to extract the details of a domain in python.

1. DNS Record

If the DNS record is empty or not found then, the value assigned to this feature is 1 or else 0

2. Website Traffic

This is the popularity of the website by determining the number of visitors and the number of pages the visitors visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information Company., 1996). If the rank of the domain < 100000, the value of this feature is 1 else 0

3. Age of the domain

Usually phishing websites live for a short period of time. So we consider the minimum age of a legitimate domain to be 6 months. Age here is nothing but different between creation and expiration time. If age of domain > 6 months, the value of this feature is 1 else 0

4. End Period od Domain

This feature can be extracted from the WHOIS database. For this feature, the remaining domain time is calculated by finding the difference between expiration time & current time.

We consider the legitimate domain is 6 months or less. If end period of the domain > 6 months, the value of this feature is 1 else 0.

The next step is to collect 4 HTML-based features. This is done with the help of various python libraries such as beautiful soup.

1. IFrame Redirection
IFrame is an HTML tag used to display an additional webpage into the currently shown page. Phishers can make use of the "iframe" tag and make the frames of the additional page invisible. If the iframe is empty or response is not found then, the value assigned to this feature is 1 or else 0.
2. Status Bar Customization
Phishers usually use JavaScript to show a fake URL in place of the status bar. To extract this feature, we must dig-out the webpage source code, particularly the "onMouseOver" event, and check if it makes any changes on the status bar. If the response is empty or onmouseover is found then, the value assigned to this feature is 1 or else 0.
3. Disabling Right Click
Phishers use JavaScript to disable the right-click function so that users cannot view and save the webpage source code. This feature is treated exactly as "Using onMouseOver to hide the Link". Nonetheless, for this feature, we will search for event "event.button==2" in the webpage source code and check if the right click is disabled. If the response is empty or onmouseover is not found then, the value assigned to this feature is 1 or else 0.
4. Website Forwarding
One of the important features that differentiates phishing websites from legitimate ones is how many times a website has been redirected. We consider that legitimate websites have been redirected one time max. Phishing websites have been redirected at least 4 times.
The final dataset as shown in figure 6 has a total of 10000 samples, 17 features and 1 label.

Analyze the dataset:
70754 Samples : 35377 phishing and 35377 legitimate samples
17 features

and One Label: 0 for legitimate and 1 for phishing URLs.

Features: ['Domain', 'Have_IP', 'Have_At', 'URL_Length', 'URL_Depth', 'Redirection', 'https_Domain', 'TinyURL', 'Prefix/Suffix', 'DNS_Record', 'Web_Traffic', 'Domain_Age', 'Domain_End', 'iFrame', 'Mouse_Over', 'Right_Click', 'Web_Forwards"]

ADDING MORE FEATURES:

17. Presence of subdomain :
Check presence of subdomain by checking '.' If present this feature is given value -1, else 1
18. Favicon:
Check the HTML script for favicon -1 if present, value = 1 if not present
19. Anchor URL :
Check presence of redirecting anchor URL If present this feature is given value -1, else 1
20. Link in Script tag:
Check links in the HTML script .-1 if present, value = 1 if not present
21. Server form handler :
Check the presence of server form handler in HTML -1 if present, value = 1 if not present
22. Info Email in HTML:
Check presence info email in the HTML script If present this feature is given value 1, else -1
23. Abnormal URL:
Check response to check abnormal URL .-1 if present, value = 1 if not present
24. Using pop up window
25. Page Rank Global page rank from www.checkpagerank.org -1 if teh rank is greater than 100000
26. Global index:
Check presence in google index If present this feature is given value 1, else -1
27. Links pointing to page: check hyperlinks in the script .-1 if present, value = 1 if not present
28. Statistical report

29. HTTPS protection
30. Request URL

So now we have the below listed features:

1. IP address
2. Length of URL
3. URL shortening
4. @ symbol
5. // Redirections
6. Suffix/Prefix '-'
7. Presence of Sub domain
8. http/https
9. End Period of domain
10. Presence of Fevicon in the script
11. Presence of non-standard port
12. HTTPS in the domain
13. Request URL
14. Anchor URL
15. Link in script tag
16. Server form handler
17. Info email in HTML
18. Abnormal URL
19. Website Forwarding
20. Status bar customization
21. Disable right click
22. Using pop up window in the script
23. IFrame redirection
24. Age of the domain
25. DNS Record
26. Website Traffic

27. Page Rank
28. Google Index
29. Links pointing to page
30. Statistical report

Details of the final Dataset:

- 70754 Samples
- 35377 phishing and 35377 legitimate samples
- 31 features
- Label: 1 for legitimate and -1 for phishing URLs
- Take 10000 data only to make the training easy.
- Analyzed the dataset.
- No outliers, No missing values.

Chapter 6

SYSTEM REQUIREMENTS

The system requirement is not that much for this project as the training has been carried out in Google Colab free version. So no specific hardware is required. The design and development of the whole system of image processing and deep learning have been carried out in Google Colab Cloud platform.

Hardware requirement:

Basic system with intel i3 or above processor.

Software requirement:

IDE used for ML development and training - Google Colab.

Language used for ML development and training - Python 3.7 - 3.11

In addition to this various python libraries like Tensorflow for deep learning are also used.
Front development in - HTML
Styling in JS.

Libraries and frameworks used:

- **Pandas** - python library for dealing with tabular data/dataframe.
- **Numpy** - python library for numerical calculations.
- **Matplotlib** - python library for plotting and visualization.
- **Re** - (regular expression) python library for dealing with strings and texts
- **Os** - python library for os-related operations, file reading and writing
- **Sci-kit learn** - python library for ML algorithms.
- **Pickle** - python library for saving high dimensional data (here, model)
- **TKinter** - Python framework to develop GUIs and python apps.
- **Whois** - library to find DNS record of URLs
- **Urllib** - Library to deal with URL strings.

- **Ipaddress** - Python IP address details library
- **BeautifulSoup** - Python library called beautiful soup for HTML related operations
- **datetime** - For the real date and time based operations
- **Requests** - python HTML requests library
- **Googlesearch librar**y -Python library for searching Google, easily.
- **Sklearn.metrics** - for ML model evaluation metrics.
- **xgboost.XGBClassifier** - For XGBoost algorithm.
- **Sklearn.metrics.accuracy_score** - for ML model evaluation metrics.
- **Seaborn library** - python library for graphs and visualizations.
- **Train_test_split - sklearn module** for splitting the dataset into train and test

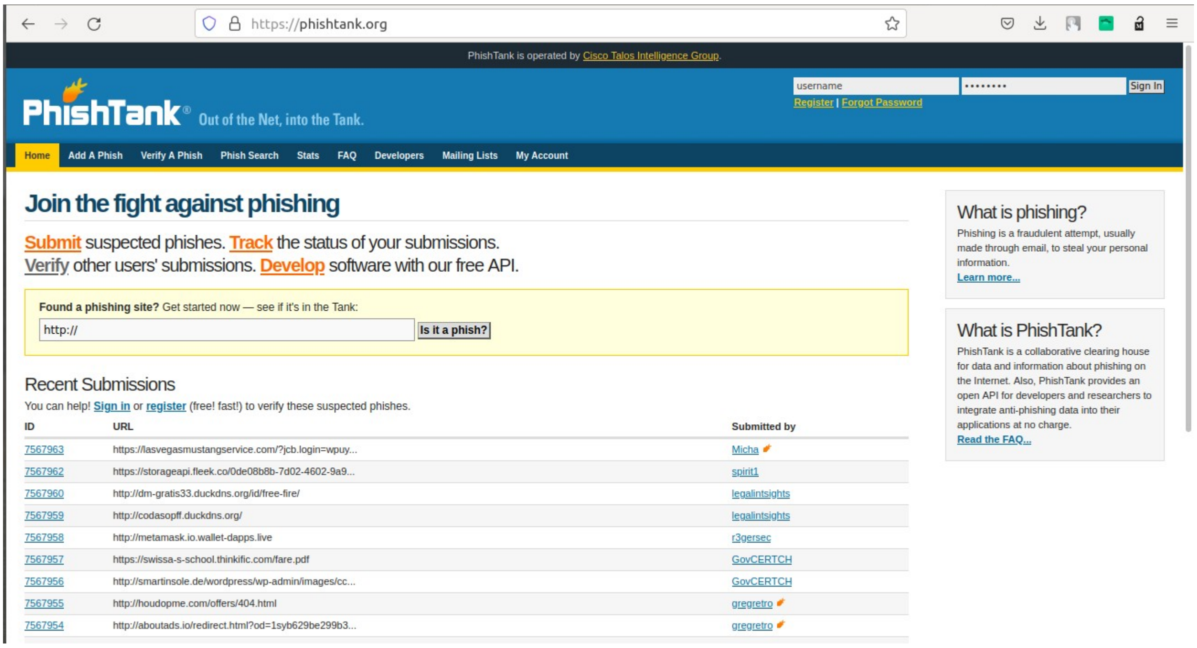
Chapter 7

MODULE DESCRIPTION

7.1 Data collection and feature extraction module.

There is a data collection and feature extraction module which is taking care of the data collection, analysis, preprocessing and feature extraction things. Then the model development module is doing all the machine learning development operations. The last one is the application module that deploys the trained model at the backend of the web browser extension.

URLs of legitimate and phishing websites are collected for ML model training. The phish URLs are collected from the open-source reservoir of www.phishtank.com. They have listed in their website, verified phishes.



Join the fight against phishing

Submit suspected phishes. Track the status of your submissions.
Verify other users' submissions. Develop software with our free API.

Found a phishing site? Get started now — see if it's in the Tank:

http://

Recent Submissions

You can help! Sign in or register (free! fast!) to verify these suspected phishes.

ID	URL	Submitted by
7567963	https://lasvegasmustangservice.com/?jcb.login=wpuy...	Micha
7567962	https://storageapi.fleek.co/0de08b8b-7d02-4602-9a9...	spirit1
7567960	http://dm-gratis33.duckdns.org/id/free-fire/	legalintsights
7567959	http://codasopff.duckdns.org/	legalintsights
7567958	http://metamask.io.wallet-dapps.live	r3gersec
7567957	https://swissa-s-school.thinkific.com/fare.pdf	GovCERTCH
7567956	http://smartinsole.de/wordpress/wp-admin/images/cc...	GovCERTCH
7567955	http://houdopme.com/offers/404.html	gregretro
7567954	http://aboutads.io/redirect.html?od=1syb629be299b3...	gregretro

What is phishing?

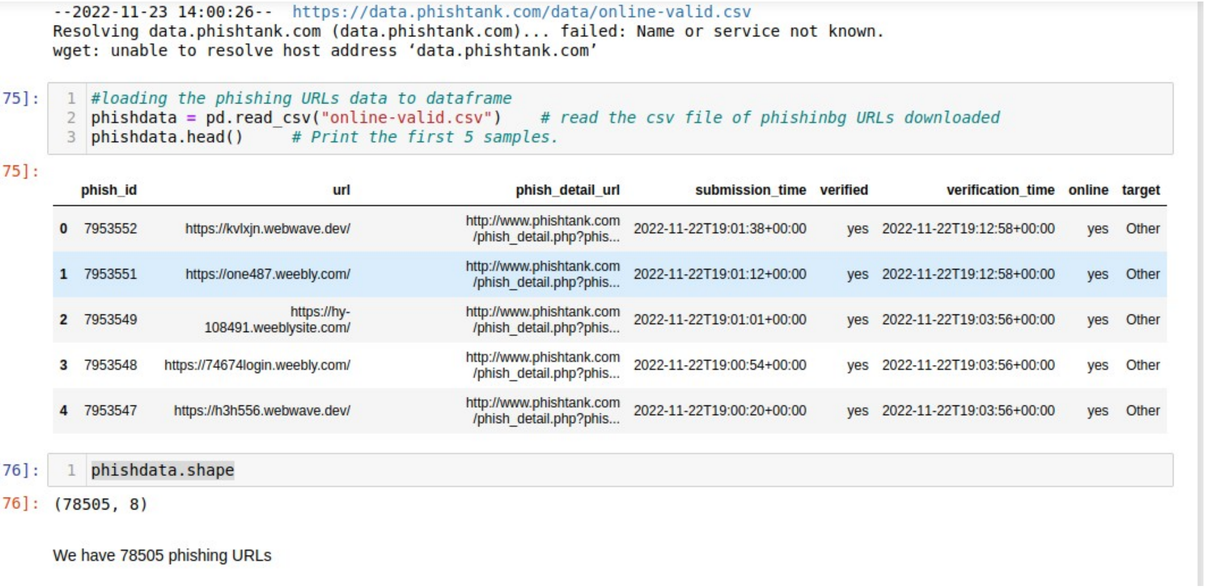
Phishing is a fraudulent attempt, usually made through email, to steal your personal information.
Learn more...

What is PhishTank?

PhishTank is a collaborative clearing house for data and information about phishing on the Internet. Also, PhishTank provides an open API for developers and researchers to integrate anti-phishing data into their applications at no charge.
Read the FAQ...

Figure 7.1 : The phishtank website where the phishing URLs are collected.

We have extracted around 78505 URLs and we restricted them into 35377 by dropping the rest. This is to make the dataset balanced with 35377 phishing and 35377 legitimate URLs.



```
--2022-11-23 14:00:26--  https://data.phishtank.com/data/online-valid.csv
Resolving data.phishtank.com (data.phishtank.com)... failed: Name or service not known.
wget: unable to resolve host address ‘data.phishtank.com’
```

```
75]:  #loading the phishing URLs data to dataframe
      phishdata = pd.read_csv("online-valid.csv")    # read the csv file of phishinbg URLs downloaded
      phishdata.head()    # Print the first 5 samples.
```

75]:

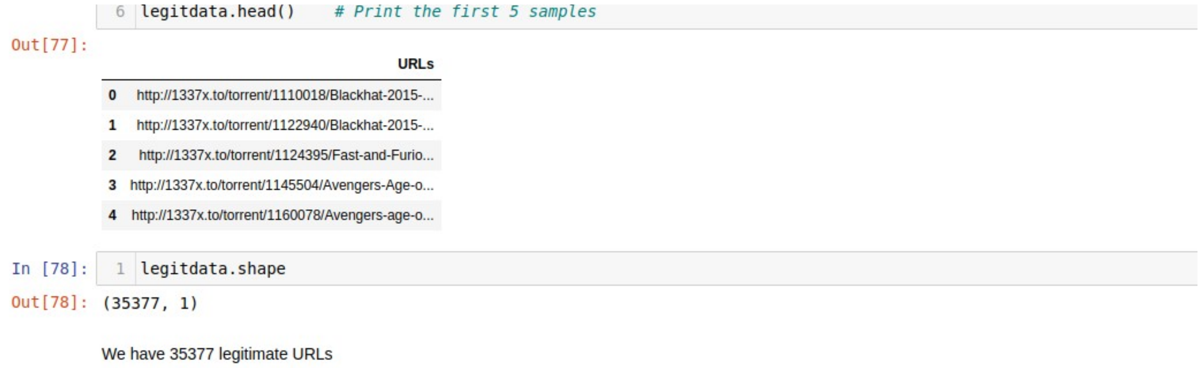
	phish_id	url	phish_detail_url	submission_time	verified	verification_time	online	target
0	7953552	https://kvlxjn.webwave.dev/	http://www.phishtank.com /phish_detail.php?phis...	2022-11-22T19:01:38+00:00	yes	2022-11-22T19:12:58+00:00	yes	Other
1	7953551	https://one487.weebly.com/	http://www.phishtank.com /phish_detail.php?phis...	2022-11-22T19:01:12+00:00	yes	2022-11-22T19:12:58+00:00	yes	Other
2	7953549	https://hy-108491.weeblysite.com/	http://www.phishtank.com /phish_detail.php?phis...	2022-11-22T19:01:01+00:00	yes	2022-11-22T19:03:56+00:00	yes	Other
3	7953548	https://74674login.weebly.com/	http://www.phishtank.com /phish_detail.php?phis...	2022-11-22T19:00:54+00:00	yes	2022-11-22T19:03:56+00:00	yes	Other
4	7953547	https://h3h556.webwave.dev/	http://www.phishtank.com /phish_detail.php?phis...	2022-11-22T19:00:20+00:00	yes	2022-11-22T19:03:56+00:00	yes	Other

```
76]:  phishdata.shape
76]:  (78505, 8)
```

We have 78505 phishing URLs

Figure 7.2: phishing URLs collected

The legitimate URLs are collected from Cyber Security research of University of New Brunswick (https://www.unb.ca/cic/datasets/url-2016.html).



```
        legitdata.head()    # Print the first 5 samples
Out[77]:
```

	URLs
0	http://1337x.to/torrent/1110018/Blackhat-2015-...
1	http://1337x.to/torrent/1122940/Blackhat-2015-...
2	http://1337x.to/torrent/1124395/Fast-and-Furio...
3	http://1337x.to/torrent/1145504/Avengers-Age-o...
4	http://1337x.to/torrent/1160078/Avengers-age-o...

```
In [78]:  legitdata.shape
Out[78]:  (35377, 1)
```

We have 35377 legitimate URLs

Figure 7.3: Legitimate URLs collected

So a total 70754 URLs are collected that have 35377 in phishing and 35377 in legitimate.

7.2 Model development module

K-NEAREST NEIGHBOR ALGORITHM (KNN)

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression problems. However, it is mainly used for classification problems in the industry.

Following are some important points regarding KNN-algorithms.

- K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique.
- KNN algorithm assumes the similarity between the new case/data and available cases and puts the new case into the category that is most similar to the available categories.
- KNN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suited category by using K- NN algorithm.
- KNN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.
- KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.

To implement the KNN algorithm we need to follow the following steps.

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of K number of neighbors

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Pros-

● Very Simple

● Training is trivial

● Works with any number of classes

● Easy to add more data

● It has few parameters such as K and distance matrix.

Cons-

● The computation cost is high because of calculating the distance between the data points for all the training samples.

● Categorical features don't work well

● Not good with the high dimensional data

DECISION TREE ALGORITHM

A decision tree is a non-parametric supervised learning algorithm, which is utilized for both classification and regression tasks. It has a hierarchical tree structure, which consists of a root node, branches, internal nodes and leaf nodes.

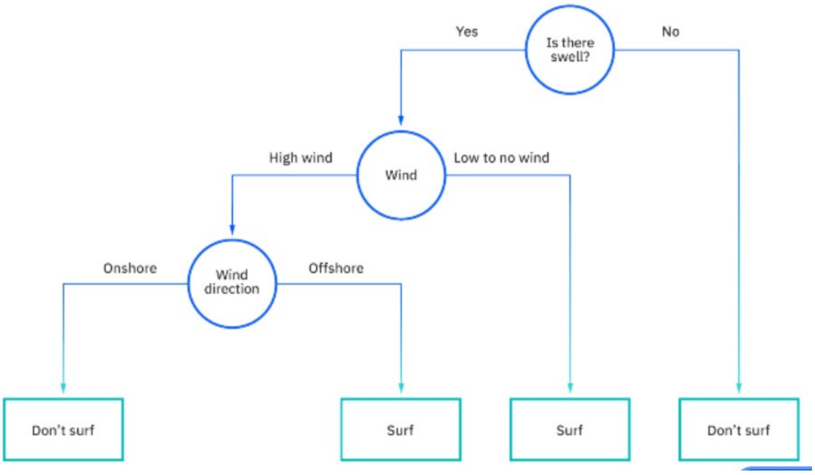


Figure 7.4: Decision Tree

As you can see from the diagram above, a decision tree starts with a root node, which does not have any incoming branches. The outgoing branches from the root node then feed into the internal nodes, also known as decision nodes. Based on the available features, both node types conduct evaluations to form homogenous subsets, which are denoted by leaf nodes, or terminal nodes. The leaf nodes represent all the possible outcomes within the dataset. As an example, let's imagine that you were trying to assess whether or not you should go surf, you may use the following decision rules to make a choice:

Types of Decision Trees Types of decision trees are based on the type of target variable we have. It can be of two types:

1. Categorical Variable Decision Tree: Decision Tree which has a categorical target variable then it is called a Categorical variable decision tree.
2. Continuous Variable Decision Tree: Decision Tree has a continuous target variable then it is called Continuous Variable Decision Tree.

Steps in the algorithm:

- It begins with the original set S as the root node.
- On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates Entropy(H) and Information gain(IG) of this attribute. It then selects the attribute which has the smallest Entropy or Largest Information gain.

- The set S is then split by the selected attribute to produce a subset of the data. The algorithm continues to recur on each subset, considering only attributes never selected before.

RANDOM FOREST ALGORITHM

Random forest is a Supervised Machine Learning Algorithm that is used widely in Classification and Regression problems. It builds decision trees on different samples and takes their majority vote for classification and average in case of regression. One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables as in the case of regression and categorical variables as in the case of classification. It performs better results for classification problems.

Real-Life Analogy Let's dive into a real-life analogy to understand this concept further. A student named X wants to choose a course after his 10+2, and he is confused about the choice of course based on his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc.
Finally, after consulting various people about the course he decides to take the course suggested by most of the people. An effective alternative is to use trees with fixed structures and random features. Tree collections are called forests, and classifiers built-in so-called random forests. The random water formation algorithm requires three arguments: the data, a desired depth of the decision trees, and a number K of the total decision trees to be built, i. The algorithm generates each of the K trees. independent, which makes it very easy to parallelize. For each tree, build a complete binary tree.

The characteristics used for the branches of this tree are selected randomly, usually with replacement, which means that the same characteristic can occur more than 20 times, even in a single branch. a. the leaves of this tree, where predictions are made, are completed based on training data. The last step is the only point at which the training data is used. The resulting classifier is just a K-lot vote, and random trees. The most amazing thing about this approach is that it actually works remarkably well. They tend to work best when all the features are at least, well, relevant, because the number of features selected for a particular tree is small. One intuitive reason that it works well is the following. Some trees will query necessary features. These trees will essentially make random predictions. But some of the trees will happen to question good characteristics and make good predictions (because the leaves are estimated based on training data).

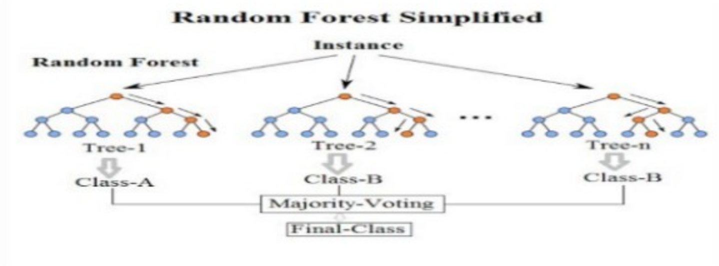


Figure 7.5 : Random forest architecture

Steps involved in random forest algorithm:

Step 1: In Random forest n number of random records are taken from the data set having k number of records.

Step 2: Individual decision trees are constructed for each sample.

Step 3: Each decision tree will generate an output.

Step 4: Final output is considered based on Majority Voting or Averaging for Classification and regression respectively.

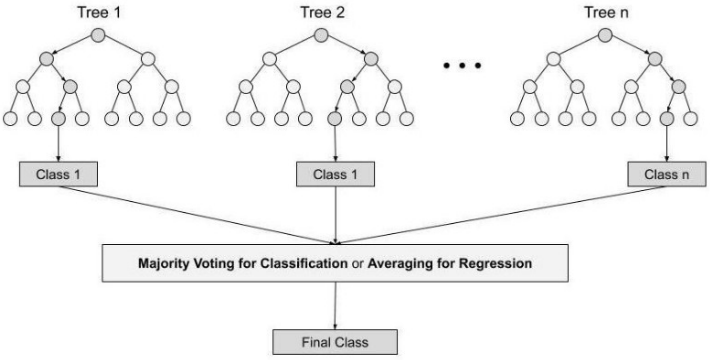

Tree 1
Tree 2
Tree n
Class 1
Class 1
Class n
Majority Voting for Classification or Averaging for Regression
Final Class

Figure 7.6: Random forest classifier procedures.

Important Hyperparameters are used in random forests to either enhance the performance and predictive power of models or to make the model faster. Following hyperparameters increases the predictive power:

1. n_estimators– number of trees the algorithm builds before averaging the predictions.
2. max_features– maximum number of features random forest considers splitting a node.
3. mini_sample_leaf– determines the minimum number of leaves required to split an internal node.

Following hyperparameters increases the speed:

1. n_jobs– it tells the engine how many processors it is allowed to use. If the value is 1, it can use only one processor but if the value is -1 there is no limit.
2. random_state– controls randomness of the sample. The model will always produce the same results if it has a definite value of random state and if it has been given the same hyperparameters and the same training data.
3. oob_score – OOB means out of the bag. It is a random forest

ARTIFICIAL NEURAL NETWORK (ANN)

The human brain, its functions, and the way it works served as the inspiration for the creation of the neural network. Artificial intelligence and machine learning, which is a subset of AI, play an essential part in its functionality. It starts working when a developer enters data and builds a machine learning algorithm, mostly using the “if ... else ...” principle of building a program. The deep neural network does not only work according to the algorithm but also can predict a solution for a task and make conclusions using its previous experience.

Nodes are little parts of the system, and they are like neurons of the human brain. When a stimulus hits them, a process takes place in these nodes. Some of them are connected and marked, and some are not, but in general, nodes are grouped into layers. The system must process layers of data between the input and output to solve a task. The more layers it has to process to get the result, the deeper the network is considered. There is a concept of Credit Assignment Path (CAP) which means the number of such layers needed for the system to complete the task. The neural network is deep if the CAP index is more than two.

At its simplest, a neural network with some level of complexity, usually at least two layers, qualifies as a deep neural network (DNN), or deep net for short. Deep nets process data in complex ways by employing sophisticated math modeling. This progress from input to output from left to right in the forward direction is called forward propagation.

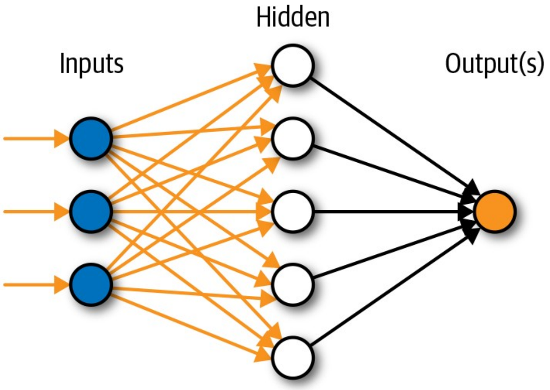


Figure 7.7: ANN

Artificial neural networks (ANNs) are biologically inspired computational networks. Among the various types of ANNs, in this chapter, we focus on multilayer perceptrons (MLPs) with backpropagation learning algorithms. MLPs, the ANNs most commonly used for a wide variety of problems, are based on a supervised procedure and comprise three layers: input, hidden, and output. We discuss various aspects of MLPs, including structure, algorithm, data preprocessing, overfitting, and sensitivity analysis. In addition, we outline the advantages and disadvantages of MLPs and recommend their usage in ecological modeling. Finally, an example demonstrating the practical application of MLP in ecological models is presented.

ANNs consist of a layer of input nodes and a layer of output nodes, connected by one or more layers of hidden nodes.

Artificial Neural Network can be best represented as a weighted directed graph, where the artificial neurons form the nodes. The association between the neurons outputs and neuron inputs can be viewed as the directed edges with weights. The Artificial Neural Network receives the input signal from the external source in the form of a pattern and image in the form of a vector. These inputs are then mathematically assigned by the notations x(n) for every n number of inputs.

Afterward, each of the input is multiplied by its corresponding weights (these weights are the details utilized by the artificial neural networks to solve a specific problem). In general terms, these weights normally represent the strength of the interconnection between neurons inside the artificial neural network. All the weighted inputs are summarized inside the computing unit. If the weighted sum is equal to zero, then bias is added to make the output non-zero or something else to scale up to the system's response. Bias has the same input, and weight equals to 1. Here the total of weighted inputs can be in the range of 0 to positive infinity. Here, to keep the response in the limits of the desired value, a certain maximum value is benchmarked, and the total of weighted inputs is passed through the activation function. The activation function refers to the set of transfer functions used to achieve the desired output. There is a different kind of the activation function, but primarily either linear or non-linear sets of functions. Some of the commonly used sets of activation functions are the Binary, linear, and Tan hyperbolic sigmoidal activation functions.

DEEP NEURAL NETWORK (DNN)

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed.

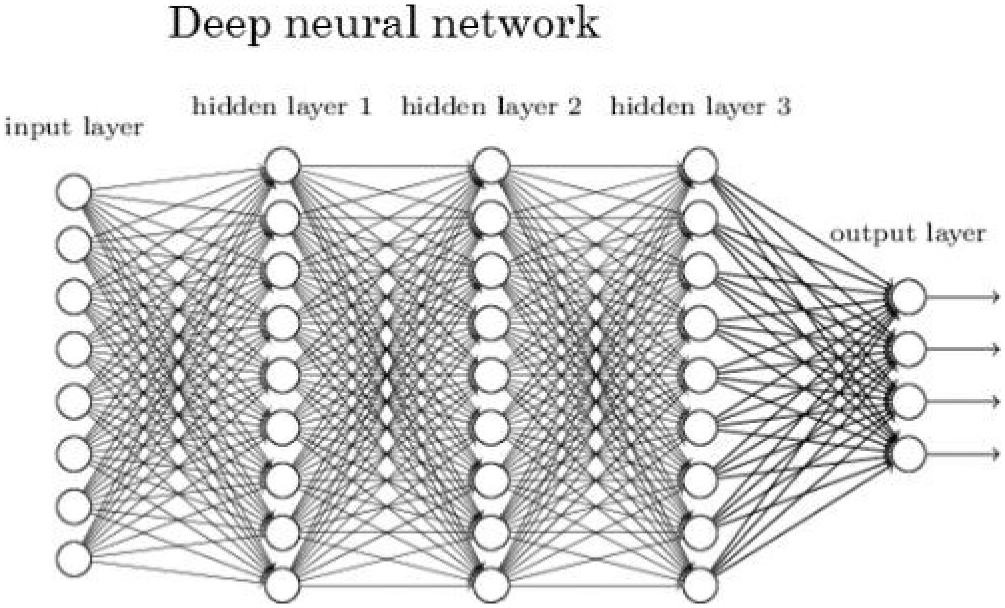


Figure 7.8: DNN

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or “weights”, to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didn’t accurately recognize a particular pattern, an algorithm would adjust the weights.That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

7.3 The Application Module

The browser extension

Thebrowser extension has been developed using HTML and CSS and front-end development languages. It has been developed with minimal GUI features to deploy the models and to detect.

Browser extensions are add-ons or plugins that you can install to have extra features or to have a better control of the web. These are powerful tools developed by thousands of software teams for different browsers. First extensions appeared in 1999. Hundreds of thousands have been made since. An extension adds features and functions to a browser. It's created using familiar web-based technologies — HTML, CSS, and JavaScript. It can take advantage of the same web APIs as JavaScript on a web page, but an extension also has access to its own set of JavaScript APIs. This means that you can do a lot more in an extension than you can with code in a web page.

Extensions can:

Enhance or complement a website: Use an add-on to deliver additional in-browser features or information from your website. Allow users to collect details from pages they visit to enhance the service you offer.

Let users show their personality: Browser extensions can manipulate the content of web pages; for example, letting users add their favorite logo or picture as a background to every page they visit. Extensions may also enable users to update the look of the Firefox UI, the same way standalone theme add-ons do.

Add or remove content from web pages: You might want to help users block intrusive ads from web pages, provide access to a travel guide whenever a country or city is mentioned in a web page, or reformat page content to offer a consistent reading experience. With the ability to access and update both a page's HTML and CSS, extensions can help users see the web the way they want to.

Add tools and new browsing features: Add new features to a taskboard, or generate QR code images from URLs, hyperlinks, or page text. With flexible UI options and the power of the WebExtensions APIs you can easily add new features to a browser. And, you can enhance almost any website's features or functionality, it doesn't have to be your website.

Add development tools: You may provide web development tools as your business or have developed a useful technique or approach to web development that you want to share. Either way, you can enhance the built-in Firefox developer tools by adding a new tab to the developer toolbar.

Development bundles associated with Google Chrome extension:

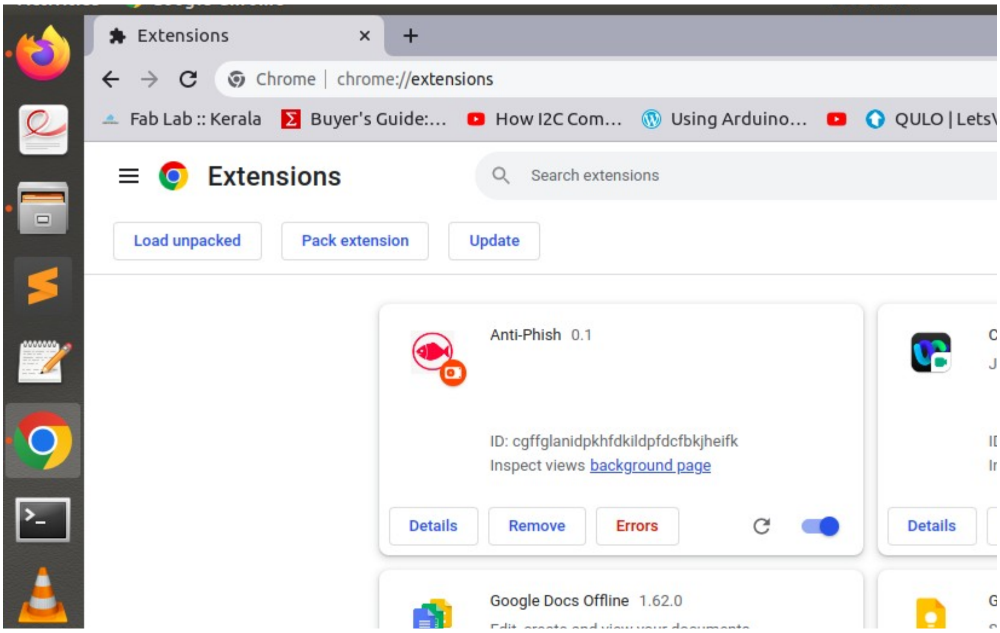
- Manifest file (manifest.json) - provided by google for giving details like name, version, details of all code files in our project, etc.

- Background script (background.js) - Tells chrome what to do like displaying result, etc.
- The User Interface (phishing .html) - For pop-up user interface.
- The content file (content.js) - For whatever happens at the backend.

Chapter 8

IMPLEMENTATION

8.1 Final Test Result.

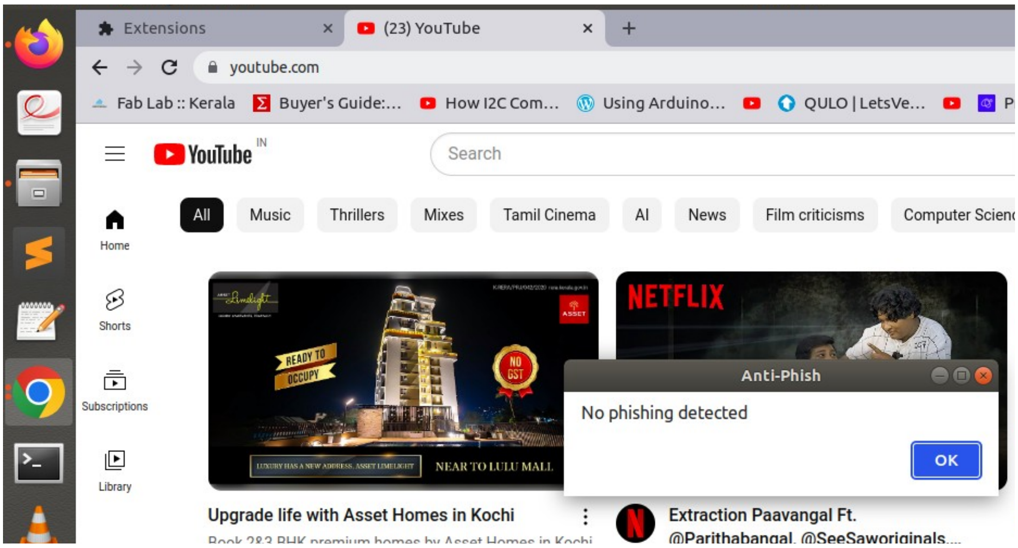


Anti-Phish 0.1

ID: cgffglanidpkhfdkildpfdcfbkjheifk

Inspect views background page

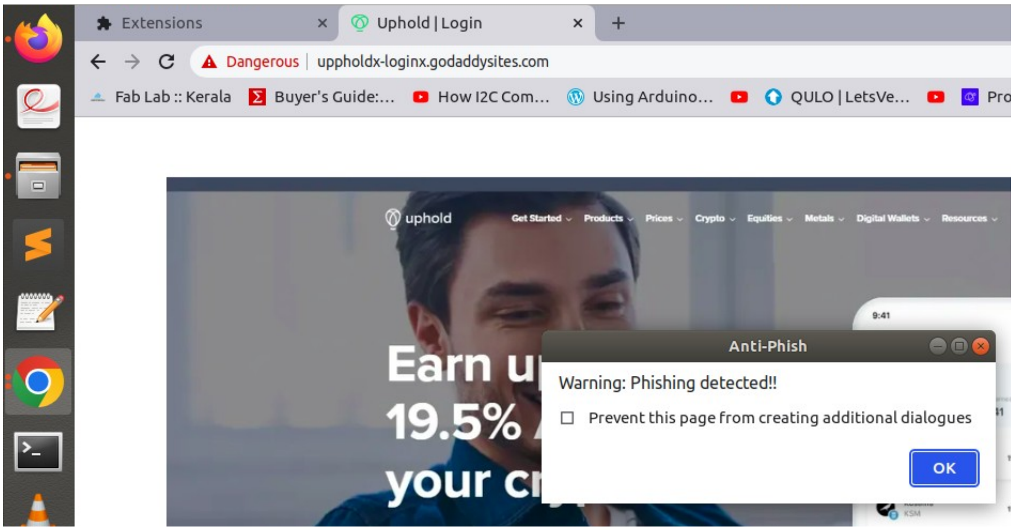
Figure 8.6 : The browser extension in chrome.



Anti-Phish

No phishing detected

Figure 8.7 : Testing a legitimate website.



Warning: Phishing detected!!

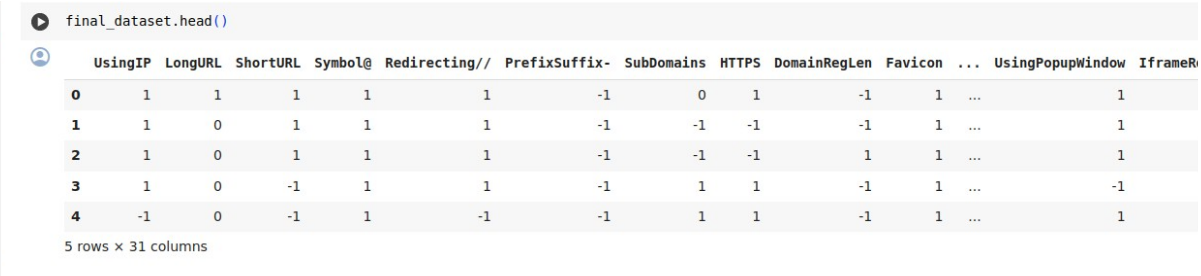
☐ Prevent this page from creating additional dialogues

Figure 8.3 : Testing a phishing website.

Chapter 9

RESULT AND ANALYSIS

9.1 Data Feature extraction.



```
final_dataset.head()
```

```
   UsingIP  LongURL  ShortURL  Symbol@  Redirecting//  PrefixSuffix-  SubDomains  HTTPS  DomainRegLen  Favicon  ...  UsingPopupWindow  IframeR
0        1        1         1        1              1             -1           0      1            -1        1  ...                 1
1        1        0         1        1              1             -1          -1     -1            -1        1  ...                 1
2        1        0         1        1              1             -1          -1     -1             1        1  ...                 1
3        1        0        -1        1              1             -1           1      1            -1        1  ...                -1
4       -1        0        -1        1             -1             -1           1      1            -1        1  ...                 1
5 rows × 31 columns
```

Figure 9.1: The final features in the dataset

9.2 ML model Training Result.

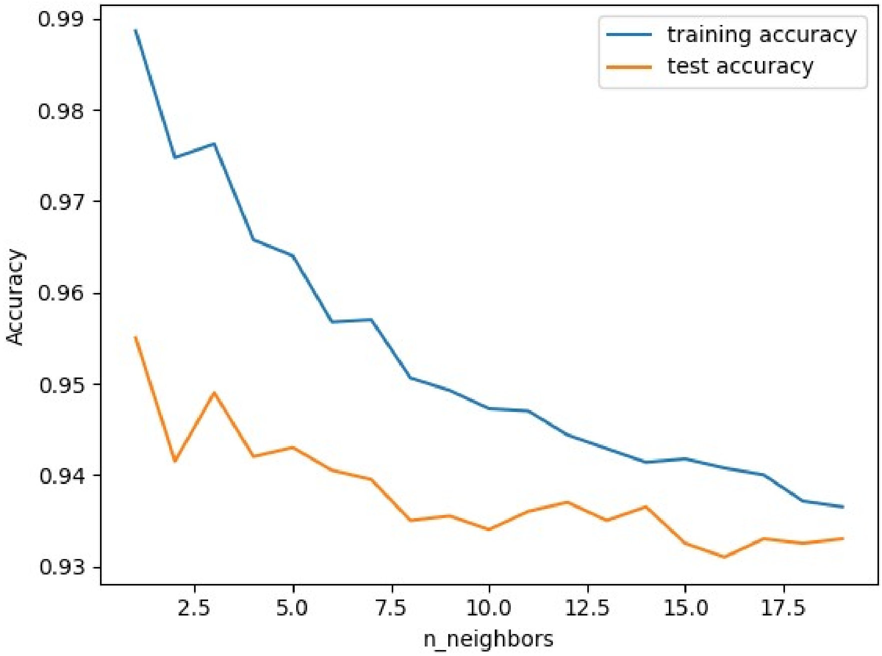


Figure 9.2: The KNN accuracy curve

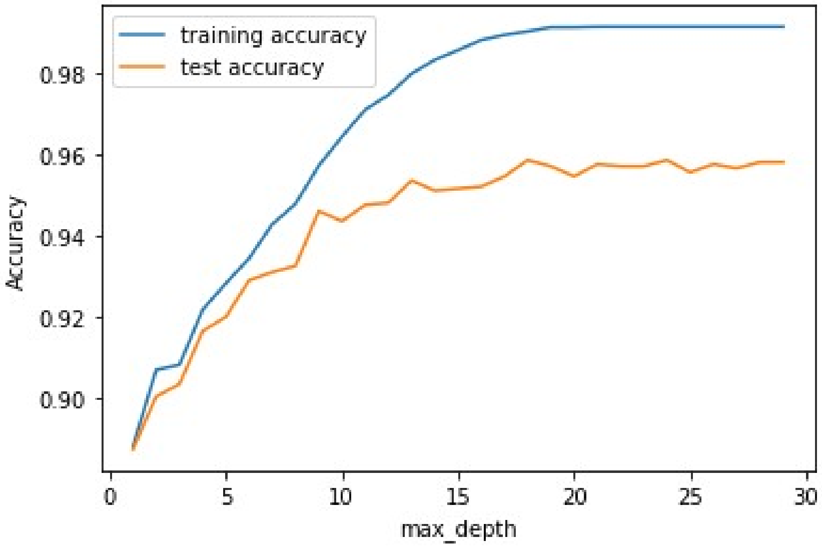


Figure 9.3: The Decision Tree accuracy curve

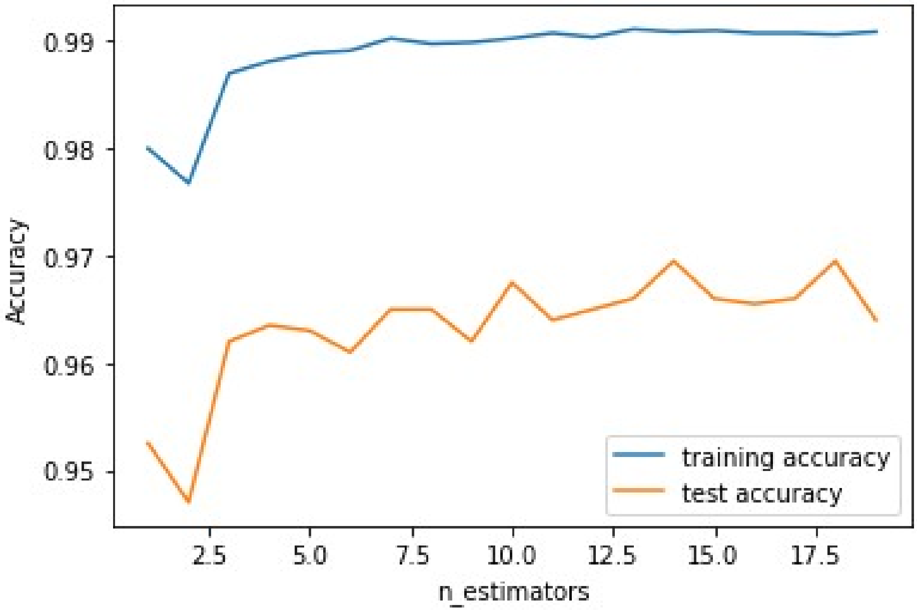
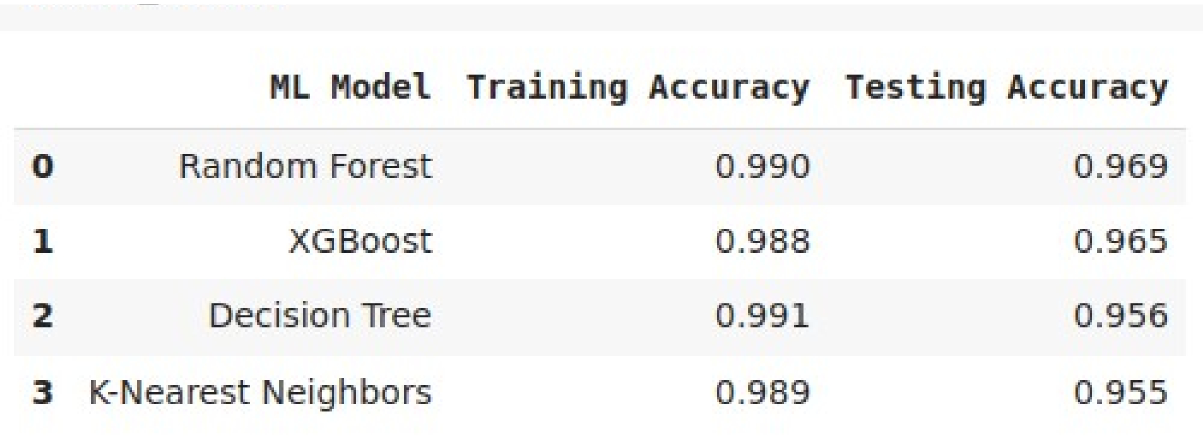


Figure 9.4: The Random Forest accuracy curve



	ML Model	Training Accuracy	Testing Accuracy
0	Random Forest	0.990	0.969
1	XGBoost	0.988	0.965
2	Decision Tree	0.991	0.956
3	K-Nearest Neighbors	0.989	0.955

Figure 9.5: ML model Training Result

9.3 DL model Training Result.

Input = 30 n vector

Hidden layer 1 = 50 neurons

Hidden layer 2 = 100 neurons

Hidden layer 3 = 100 neurons

Output layer = 2 neurons

DNN Classifier on Train data: Accuracy - 99.1

DNN Classifier on Test data: Accuracy - 96.1

Chapter 10
CONCLUSION

Website phishing is the biggest threat in the cyber security space! Yet, there is not much defense system present that has the characteristics such as high accuracy and light weight. We collected data for both the problems and carried out an extensive preprocessing on it to make it a machine learning trainable dataset. Machine Learning models such as KNN, Decision Tree, and Random forest, XGBoost, etc are trained. After testing and evaluating with performance evaluation metrics, Random Forest model found to be performing well and saved the models as pickle files. Developed a Google Chrome extension and deployed the model and tested it for phishing detection in real-time. As a future thought, the model developed may be deployed as a callable open source python API so that anyone can access it anytime. Still, the accuracy of the model can further be increased by adding more training data and with more training hardware resources.

REFERENCES

1. Ankit Kumar Jain & B. B. Gupta, 2018. "Towards detection of phishing websites on client-side using machine learning based approach," Telecommunication Systems: Modelling, Analysis, Design and Management, Springer, vol. 68(4), pages 687-700, August.

2. Dutta, Ashit. (2021). Detecting phishing websites using machine learning technique. PloS one. 16. e0258361. 10.1371/journal.pone.0258361.

3. M. Abutaha, M. Ababneh, K. Mahmoud and S. A. -H. Baddar, "URL Phishing Detection using Machine Learning Techniques based on URLs Lexical Analysis," 2021 12th International Conference on Information and Communication Systems (ICICS), 2021, pp. 147-152, doi: 10.1109/ICICS52457.2021.9464539.

4. S. Alrefaai, G. Özdemir and A. Mohamed, "Detecting Phishing Websites Using Machine Learning," 2022 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), 2022, pp. 1-6, doi: 10.1109/HORA55278.2022.9799917.

5. B. Geyik, K. Erensoy and E. Kocyigit, "Detection of Phishing Websites from URLs by using Classification Techniques on WEKA," 2021 6th International Conference on Inventive Computation Technologies (ICICT), 2021, pp. 120-125, doi: 10.1109/ICICT50816.2021.9358642.

6. S. Singh, M. P. Singh and R. Pandey, "Phishing Detection from URLs Using Deep Learning Approach," 2020 5th International Conference on Computing, Communication and Security (ICCCS), 2020, pp. 1-4, doi: 10.1109/ICCCS49678.2020.9277459.

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

8. S. Parekh, D. Parikh, S. Kotak and S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 949-952, doi: 10.1109/ICICCT.2018.8473085.

9. H. Yuan, X. Chen, Y. Li, Z. Yang and W. Liu, "Detecting Phishing Websites and Targets Based on URLs and Webpage Links," 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 3669-3674, doi: 10.1109/ICPR.2018.8546262.

10. M. N. Feroz and S. Mengel, "Phishing URL Detection Using URL Ranking," 2015 IEEE International Congress on Big Data, 2015, pp. 635-638, doi: 10.1109/BigDataCongress.2015.97.

11. C. L. Tan, K. L. Chiew and S. N. Sze, "Phishing website detection using URL-assisted brand name weighting system," 2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), 2014, pp. 054-059, doi: 10.1109/ISPACS.2014.7024424.

12. Y. Su, "Research on Website Phishing Detection Based on LSTM RNN," 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), Chongqing, China, 2020, pp. 284-288, doi: 10.1109/ITNEC48623.2020.9084799.

13. Detection of phishing URLs using machine learning techniques. Available from:https://www.researchgate.net/publication/269032183_Detection_of_phishing_URLs_using_machine_learning_techniques [accessed Jun 27 2022].

14. Masum, Mohammad & Hossain Faruk, Md Jobair & Shahriar, Hossain & Qian, Kai & Lo, Dan & Adnan, Muhaiminul. (2022). Ransomware Classification and Detection With Machine Learning Algorithms. 10.1109/CCWC54503.2022.9720869.

15. Abu Saad Choudhary , Rucha Desai , Lavkush Gupta , Madhuri Gedam “Detection and Prevention of Phishing Attacks” Asian Journal of Convergence in Technology.

16. Narendra. M. Shekokar, Chaitali Shah, Mrunal Mahajan,Shruti Rachh “An ideal approach for detection and prevention of phishing attacks”

17. P. Vimala Manohara Ruth, Y. Rama Devi, Haritha, Shiva Kumar “Prediction of Phishing Website for Data Security Using Various Machine Learning Algorithms”

18. Rishikesh Mahajan and Irfan Siddavatam “Phishing Website Detection using Machine Learning Algorithms”

19. Narendra. M. Shekokar, Chaitali Shah, Mrunal Mahajan,Shruti Rachh “An Ideal Approach For Detection And Prevention of Phishing Attacks”

20. Mahmoud Khonji, Youssef Iraqi, Senior Member, IEEE, and Andrew Jones “Phishing Detection: A Literature Survey”.

Appendix.

1 Model Training (XGBoost).

```
!pip install xgboost==1.5
```

```
#  XGBoost Classifier Model
from xgboost import XGBClassifier
```

```
# instantiate the model
xgb = XGBClassifier()
# fit the model
xgb.fit(X_train,y_train)
```

```
#predicting the target value from the model for the samples
y_train_xgb = xgb.predict(X_train)
y_test_xgb = xgb.predict(X_test)
```

```
#computing the accuracy, f1_score, Recall, precision of the model
performance

acc_train_xgb = metrics.accuracy_score(y_train,y_train_xgb)
acc_test_xgb = metrics.accuracy_score(y_test,y_test_xgb)
print("XGBoost Classifier : Accuracy on training Data:
{:.3f}".format(acc_train_xgb))
print("XGBoost Classifier : Accuracy on test Data:
{:.3f}".format(acc_test_xgb))
```

```
# Save the model

filename = "XGBoost_model.pickle"

# save model
pickle.dump(knn, open(filename, "wb"))
```

2 Testing URL.

```
# Import the feature extraction as a full python file

from feature import generate_data_set
import pickle
import numpy as np
```

```
# Make single function

def Test_URL():
 URL =  input('Enter the URL: ')
 Model = pickle.load(open("XGBoost_model.pickle", "rb"))
  x  =  np.array(generate_data_set(URL)).reshape(1,30)      #  The  URL  is
converted into ML model's dataset format by using feature extractoion
 y_pred = Model.predict(x)[0]    # Pass the data to the model to detect
 if y_pred== -1:
       result = ' ---> This is a phishing URL!'

 elif y_pred==1:
       result = ' ---> This is a legitimate URL'

 print(result)
```

```
Test_URL()
```

Enter the URL: http://www.facebook.com

 ---> This is a legitimate URL

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but KNeighborsClassifier was fitted with feature names

  warnings.warn()