PHISHING WEBSITE DETECTION USING NOVEL MACHINE LEARNING FUSION APPROACH

Submitted in partial fulfillment of the requirements
for the award of Bachelor of Engineering degree
in Computer Science and Engineering

By

Arikatla gopi Venkata Sudheer(38110046)
Aravapalli Sujith Kumar(38110044)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with Grade "A" by NAAC

JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600 119

MARCH – 2022

SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

Accredited with “A” grade by NAAC

Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600119

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Aravapalli sujith kumar (38110044)** and **Arikatla gopi venkatasudheer (38110046)** who carried out the project entitled “**PHISHING WEBSITE DETECTION USING NOVEL MACHINE LEARNING FUSION APPROACH”** under my supervision from November 2021 to March2022.

Internal Guide

Dr. M. MAHESHWARI M.E., Ph.D.

Head of the Department

Dr. S. Vigneshwari M.E., Ph.D., and Dr. L. Lakshmanan M.E., Ph.D.,

---

Submitted for Viva voce Examination heldon_____

InternalExaminer **ExternalExaminer**

DECLARATION

I **ARAVAPALLI SUJITH KUMAR (Reg No:38110044) and ARIKATLA GOPI VENKATA SUDHEER (Reg No: 38110046)** hereby declare that the Project Report entitled "**PHISHING WEBSITE DETECTION USING NOVEL MACHINE LEARNING FUSION APPROACH**" done by us under the guidance of **Dr. B.MAHESHWARI M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in 2018-2022.

DATE:

PLACE: **SIGNATURE OF THE CANDIDATE**

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management** of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E.,Ph.D.**, **Dean**, School of Computing **Dr.S.Vigneshwari M.E., Ph.D.** and **Dr.L.Lakshmanan M.E., Ph.D. ,** Heads of the Department of Computer Science and Engineering for providing us necessary support and details at the right time during the progressivereviews.

I would like to express my sincere and a deep sense of gratitude to my Project Guide **Dr.M.MaheshwariM.E.,Ph.D.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

TABLE OF CONTENTS

ABSTARCT:

Phishing websites have proven to be a major security concern. Several cyber attacks risk the confidentiality, integrity, and availability of company and consumer data, and phishing is the beginning point for many of them.Many researchers have spent decades creating unique approaches to automatically detect phishing websites. While cutting-edge solutions can deliver better results, they need a lot of manual feature engineering and aren't good at identifying new phishing attacks. As a result, finding strategies that can automatically detect phishing websites and quickly manage zero-day phishing attempts is an open challenge in this field. The web page in the URL which hosts that contains a wealth of data that can be used to determine the web server's maliciousness.Machine Learning is an effective method for detecting phishing.It also eliminates the disadvantages of the previous method.We conducted a thorough review of the literature and suggested a new method for detecting phishing websites using features extraction and a machine learning algorithm. The goal of this research is to use the dataset collected to train ML models and deep neural nets to anticipate phishing websites.

Chapter 1

INTRODUCTION:

Phishing is the most unsafe criminal exercises in cyber space. Since most of the users go online to access the services provided by government and financial institutions, there has been a significant increase in phishing attacks for the past few years. Phishers started to earn money and they are doing this as a successful business. Various methods are used by phishers to attack the vulnerable users such as messaging, VOIP, spoofed link and counterfeit websites. It is very easy to create counterfeit websites, which looks like a genuine website in terms of layout and content. Even, the content of these websites would be identical to their legitimate websites. The reason for creating these websites is to get private data from users like account numbers, login id, passwords of debit and credit card, etc. Moreover, attackers ask security questions to answer to posing as a high level security measure providing to users. When users respond to those questions, they get easily trapped into phishing attacks. Many researches have been going on to prevent phishing attacks by different communities around the world. Phishing attacks can be prevented by detecting the websites and creating awareness to users to identify the phishing websites. Machine learning algorithms have been one of the powerful techniques in detecting phishing websites. In this study, various methods of detecting phishing websites have been discussed.

Chapter 2

Literature review

2.1 Detection of Phishing URL using Machine Learning

Abstract:

Phishing websites have proven to be a major security concern. Several cyberattacks
risk the confidentiality, integrity, and availability of company and consumer data, and phishing is the beginning point for many of them. Many researchers have spent decades creating unique approaches to automatically detect phishing websites. While cutting-edge solutions can deliver better results, they need a lot of manual feature engineering and aren't good at identifying new phishing attacks. As a result, finding strategies that can automatically detect phishing websites and quickly manage zero-day phishing attempts is an open challenge in this field. The web page in the URL which hosts that contains a wealth of data that can be used to determine the web server's maliciousness. Machine Learning is an effective method for detecting phishing.It also eliminates the disadvantages of the previous method. We conducted a thorough review of the literature and suggested a new method for detecting phishing websites using features extraction and a machine learning algorithm. The goal of this research is to use the dataset collected to train ML models and deep neural nets to anticipate phishing websites.

INTRODUCTION

Phishing has become the most serious problem, harming individuals, corporations, and even entire countries. The availability of multiple services such as online banking, entertainment, education, software downloading, and social networking has accelerated the Web's evolution in recent years. As a result, a massive amount of data is constantly downloaded and transferred to the Internet. Spoofed e-mails pretending to be from reputable businesses and agencies are used in social engineering techniques to direct consumers to fake websites that deceive users into giving financial information such as usernames and passwords. Technical tricks involve the installation of malicious software on computers to steal credentials directly, with systems frequently used to intercept users' online account usernames and passwords.

A. Types of Phishing Attacks

• ***Deceptive Phishing****:*

This is the most frequent type of phishing assault, in which a

Cyber criminal impersonates a well-known institution, domain, or organization to acquire sensitive personal information from the victim, such as login credentials, passwords, bank account information, credit card information, and so on. Because there is no personalization or customization for the people, this form of attack lacks sophistication.

•***Spear Phishing***: Emails containing malicious URLs in this sort of phishing email contain a lot of personalization information about the potential victim. The recipient's name, company name, designation, friends, co-workers, and other social information may be included in the email.

•***Whale Phishing***: To spear phish a "whale," here a top-level executive such as CEO, this sort of phishing targets corporate leaders such as CEOs and top-level management employees.

• ***URL Phishing***: To infect the target, the fraudster or cyber-criminal employs a URL link. People are sociable creatures who will eagerly click the link to accept friend invitations and may even be willing to disclose personal information such as email addresses.

This is because the phishers are redirecting users to a false web server. Secure browser connections are also used by attackers to carry out their unlawful actions. Due to a lack of appropriate tools for combating phishing attacks, firms are unable to train their staff in this area, resulting in an increase in phishing attacks.Companies are educating their staff with mock phishing assaults, updating all their systems with the latest security procedures, and encrypting important Information as broad countermeasures. Browsing without caution is one of the most common ways to become a victim of this phishing assault. The appearance of phishing websites is like that of authentic websites.

***Research question*:**

Are some of the research questions on which this research paper will elaborate.

• Is it possible to extract features from the URL using machine learning techniques?

• How can phishing URLs be detected using a Machine learning approach in terms of efficiency?

The ultimate purpose of this study work is to provide a better understanding of the process of identifying the presence of Phishing attacks using a machine learning technique to identify URL based features like Address Bar, Domain, JavaScript, and HTML based features. The remaining part of the paper is written out as follows. The Section 2 of paper is dedicated to a literature review.

Section 3 outlines the planned research approach, Section 4 presents the experimental data, and Section 5 provides the conclusion.

Literature Review

Many scholars have done some sort of analysis on the statistics of phishing URLs. Our technique incorporates key concepts from past research. We review past work in the detection of phishing sites using URL features, which inspired our current approach. Happy describe phishing as "one of the most dangerous ways for hackers to obtain users' accounts such as usernames, account numbers and passwords, without their awareness." Users are ignorant of this type of trap and will ultimately, they fall into Phishing scam. This could be due to a lack of a combination of financial aid and personal experience, as well as a lack of market awareness or brand trust. In this article, Mehmet et al. suggested a method for phishing detection based on URLs. To compare the results, the researchers utilized eight different algorithms to evaluate the URLs of three separate datasets using various sorts of machine learning methods and hierarchical architectures. The first method evaluates various features of the URL; the second method investigates the website's authenticity by determining where it is hosted and who operates it; and the third method investigates the website's graphic presence.We employ Machine Learning techniques and algorithms to analyse these many properties of URLs and websites. Garera et al. classify phishing URLs using logistic regression over hand-selected variables. The inclusion of red flag keywords in the URL, as well as features based on Google's Web page and Google's Page Rank quality recommendations, are among the features. Without access to the same URLs and features as our approach, it's difficult to conduct a direct comparison. In this research, Yong et al. created a novel approach for detecting phishing websites that focuses on detecting a URL which has been demonstrated to be an accurate and efficient way of detection. To offer you a better idea, our new capsule-based neural network is divided into several parallel components. One method involves removing shallow characteristics from URLs. The other two, on the other hand, construct accurate feature representations of URLs and use shallow features to evaluate URL legitimacy. The final output of our system is calculated by adding the outputs of all divisions. Extensive testing on a dataset collected from the Internet indicate that our system can compete with other cutting-edge detection methods while consuming a fair amount of time. For phishing detection, Vahid Shahrivari et al. used machine learning approaches. They used the logistic regression classification method, KNN,

Adaboost algorithm, SVM, ANN and random forest. They found random forest algorithm provided good accuracy. Dr.G. Ravi Kumar used a variety of machine learning methods to detect phishing assaults. For improved results, they used NLP tools. They were able to achieve high accuracy using a Support Vector Machine and data that had been pre-processed using NLP approaches. Amani Alswailem et al. tried different machine learning model for phishing detection but was able to achieve more accuracy in random forest. Hossein et al. created the “Fresh-Phish” open-source framework. This system can be used to build machine-learning data for phishing websites. They used a smaller feature set and built the query in Python. They create a big, labelled dataset and test several machine-learning classifiers on it. Using machine-learning classifiers, this analysis yields very high accuracy. These studies look at how long it takes to train a model. X. Zhang suggested a phishing detection model based on mining the semantic characteristics of word embedding, semantic feature, and multi-scale statistical features in Chinese web pages to detect phishing performance successfully. To obtain statistical aspects of web pages, eleven features were retrieved and divided into five classes. To obtain statistical aspects of web pages, eleven features were retrieved and divided into five classes. To learn and evaluate the model, AdaBoost, Bagging, Random Forest, and SMO are utilized. The legitimate URLs dataset came from DirectIndustry online guides, and the phishing data came from China's Anti-Phishing Alliance. With novel methodologies, M. Aydin approaches a framework for extracting characteristics that is versatile and straightforward. Phish Tank provides data, and Google provides authentic URLs. C# programming and R programming were utilized to btain the text attributes. The dataset and third-party service providers yielded a total of 133 features. The feature selection approaches of CFS subset based and Consistency subset-based feature selection were employed and examined with the WEKA tool. The performance of the Nave Bayes and Sequential Minimal Optimization (SMO) algorithms was evaluated, and the author prefers SMO to NB for phishing detection.

Research Methodology

A phishing website is a social engineering technique that imitates legitimate webpages and uniform resource locators (URLs). The Uniform Resource Locator (URL) is the most common way for phishing assaults to occur. Phisher has complete control over the URL's sub-domains. The phisher can alter the URL because it contains file components and directories.

Methodologies

This research used the linear-sequential model, often known as the waterfall model. Although the waterfall approach is considered conventional, it works best in instances where there are few requirements. The application was divided into smaller components that were built using frameworks and hand-written code.

Research Framework:

The steps of this research in which some selected publications were read to determine the research gap and, as a result, the research challenge was defined. Feature selection, classification and phishing website detection were all given significant consideration. It's worth noting that most phishing detection researchers rely on datasets they've created. However, because the datasets utilized were not available online for those who use and check their results, it is difficult to assess and compare the performance of a model with other models. As a result, such results cannot be generalized.

Language

For the preparation of this dissertation, I used Python as the primary language. Python is a language that is heavily focused on machine learning. It includes several machine learning libraries that may be utilized straight from an import. Python is commonly used by developers all around the world to deal with machine learning because of its extensive library of machine learning libraries. Python has a strong community, and as a result, new features are added with each release.

Data Collection

The phishing URLs were gathered using the open source tool Phish Tank. This site provides a set of phishing URLs in a variety of forms, including csv, json, and others, which are updated hourly. This dataset is used to train machine learning models with 5000 random phishing URLs.

Data Cleaning

Fill in missing numbers, smooth out creaking data, detect and delete outliers, and repair anomalies to clean up the data.

Data Pre-processing

Data pre-processing is a cleaning operation that converts unstructured raw data into a neat, well-structured dataset that may be used for further research. Data pre-processing is a cleaning operation that transforms unstructured raw data into well-structured and neat dataset which can be used for further research.

Extraction of Features

In the literature and commercial products, there are numerous algorithms and data formats for phishing URL detection. A phishing URL and its accompanying website have various characteristics that distinguish them from harmful URLs. For example, to mask the true domain name, an attacker can create a long and complicated domain name. Different types of features that are used in machine learning algorithms in the academic study detection process are used. The following is a list of features gathered from academic studies for phishing domain detection using machine learning approaches. Because of some constraints, it may not be logical to use some of the features in specific instances. Using Content-Based Features to construct a quick detection mechanism capable of analyzing a huge number of domains may not be feasible. Page-Based Features are not very effective when analyzing registered domains. As a result, the features that the detection mechanism will use are determined by the detection mechanism's purpose. So, which features should be used in the detecting technique been carefully chosen.

Models and Training

The data is split into 8000 training samples and 2000 testing samples, before the ML model is trained. It is evident from the dataset that this is a supervised machine learning problem. Classification and regression are the two main types of supervised machine learning issues. Because the input URL is classed as legitimate or phishing, this data set has a classification problem. The following supervised machine learning models were examined for this project's dataset training:

- Decision Tree
- Multilayer Perceptron

• Random Forest
• Autoencoder Neural Network
• XGBoost
• Support Vector Machines

***Design Specification*:**

The project is having three features that been extracted from data. The features are Address Bar based, Domain based, and HTML and JavaScript based. In the below section will discuss in detail.

Address based

Below are the categories been extracted from address based

1. Domain of the URL

Where domain which is present in the URL been extracted

2. IP Address in the URL

The presence of an IP address in the URL is checked. Instead of a domain name, URLs may contain an IP address. If an IP address is used instead of a domain name in a URL, we can be certain that the URL is being used to collect sensitive information.

3. "@" Symbol in URL

The presence of the'@' symbol in the URL is checked. When the “@” symbol is used in a URL, the browser ignores anything before the “@” symbol, and the genuine address is commonly found after the “@” symbol.

4. Length of URL

Calculates the URL's length. Phishers can disguise the suspicious element of a URL in the address bar by using a lengthy URL. If the length of the URL is larger than or equal to 54 characters, the URL is classed as phishing in this project.

5. Depth of URL

Calculates the URL's depth. Based on the'/', this feature determines the number of subpages in the given address.

6. Redirection "//" in URL

The existence of"//" in the URL is checked. The presence of the character"//" in the URL route indicates that the user will be redirected to another website. The position of the"//" in the URL is

calculated. We discovered that if the URL begins with “HTTP,” the “//” should be placed in the sixth position. If the URL uses “HTTPS,” however, the “//” should occur in the seventh place.

7. Http/Https in Domain name

The existence of "http/https" in the domain part of the URL is checked. To deceive users, phishers may append the “HTTPS” token to the domain section of a URL.

8. Using URL Shortening Services

URL shortening is a means of reducing the length of a URL while still directing to the desired webpage on the "World Wide Web." This is performed by using a “HTTP Redirect” on a short domain name that points to a webpage with a long URL.

9. Prefix or Suffix "-" in Domain

Checking for the presence of a '-' in the URL's domain part. In genuine URLs, the dash symbol is rarely used. Phishers frequently append prefixes or suffixes to domain names, separated by (-), to give the impression that they are dealing with a legitimate website.

Domain based

This category contains a lot of features that can be extracted. This category contains a lot of features that can be extracted. The following were considered for this project out of all of them.

1. DNS Record

In the case of phishing websites, the WHOIS database either does not recognize the stated identity or there are no records for the host name .

2. Web Traffic

This function determines the number of visitors and the number of pages they visit to determine the popularity of the website. In the worst-case circumstances, legitimate websites placed among the top100,000, according to our data. Furthermore, it is categorized as "Phishing" if the domain has no traffic or is not recognized by the Alexa database.

3. Age of Domain

This information can be retrieved from the WHOIS database. Most phishing websites are only active for a short time. For this project, the minimum age of a legal domain is deemed to be 12 months. Age is simply the difference between the time of creation and the time of expiry.

4. End Period of Domain

This information can be gleaned from the WHOIS database. The remaining domain time is calculated for this feature by determining the difference between the expiry time and the current time. For this project, the valid domain's end time is regarded to be 6 months or fewer.

HTML and JavaScript based

Many elements that fall within this group can be extracted. The following were considered for this project out of all of them.

1. IFrame Redirection

IFrame is an HTML tag that allows you to insert another webpage into the one you're now viewing. The “iframe” tag can be used by phishers to make the frame invisible, i.e., without frame borders. Phishers employ the “frame border” attribute in this case, which causes the browser to create a visual boundary.

2. Status Bar Customization

Phishers may utilize JavaScript to trick visitors into seeing a false URL in the status bar. To get this feature, we'll need to delve into the webpage source code, specifically the "on Mouseover" event, and see if it alters the status bar.

3. Disabling Right Click

Phishers disable the right-click function with JavaScript, preventing users from viewing and saving the webpage source code. This functionality is handled in the same way as "Hiding the Link with on Mouseover." Nonetheless, we'll look for the event "event. button==2" in the webpage source code and see if the right click is disabled for this functionality.

4. Website Forwarding

The number of times a website has been redirected is a narrow line that separates phishing websites from authentic ones. We discovered that authentic websites were only routed once in our sample. Phishing websites with this functionality, on the other hand, have been redirected at least four times.

5. Implementation

We'll examine at the implementation component of our artefact in this area of the report, with a focus on the description of the developed solution. This is a task that requires supervised machine learning.

Dataset

We collected the datasets from the open-source platform called Phishing tank. The dataset that collected was in csv format. There are 18 columns in the dataset, and we transformed the dataset by applying data pre-processing technique. To see the features in the data we used few of the data frame methods for familiarizing. For visualization, and to see how the data is distributed and how features are related to one another, a few plots and graphs are given. The Domain column has no bearing on the training of a machine learning model. We now have 16 features and a target column. The recovered features of the legitimate and phishing URL datasets are simply concatenated in the feature extraction file, with no shuffling. We need to shuffle the
data to balance out the distribution while breaking it into training and testing sets. This also eliminates the possibility of over fitting during model training.

Machine Learning Models

For phishing website identification, we used many machine learning methods. We used the classification and regression algorithms listed below.

Decision Tree Classifier

For classification and regression applications, decision trees are commonly used models. They basically learn a hierarchy of if/else questions that leads to a choice. Learning a decision tree is memorizing the sequence of if/else questions that leads to the correct answer in the shortest amount of time. The method runs through all potential tests to discover the one that is most informative about the target variable to build a tree.

Random Forest Classifier

Random forests are one of the most extensively used machine learning approaches for regression and classification. A random forest is just a collection of decision trees, each somewhat different from the others. The notion behind random forests is that while each tree may do a decent job of predicting, it will almost certainly overfit on some data. They are incredibly powerful, frequently operate effectively without a lot of parameters adjusting, and don't require data scalability.

MLPs

Feed-forward neural networks, or simply neural networks, are another name for multilayer perceptron's. MLPs are expansions of linear models that conduct many steps of processing to arrive at a decision. They can be used for both classification and regression problems.

XGBoost

These days, XGBoost is one of the most prominent machine learning algorithms. eXtreme Gradient Boosting is the abbreviation for XGBoost. Regardless of whether the goal at hand regression or classification is, XGBoost is a high-performance and high-speed implementation of gradient boosted decision trees.

Autoencoder

A neural network with the same number of input neurons as output neurons is known as an auto encoder. The input/output neurons will have fewer neurons than the hidden layers of the neural network. The auto-encoder must learn to encode the input to the fewer hidden neurons since there are less neurons. In an auto encoder, the predictors (x) and output (y) are identical.

SVM

SVM are supervised learning models with related learning algorithms used in machine learning to examine data for classification and regression analysis. An SVM training algorithm creates a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples that are individually designated as belonging to one of two categories.

Environmental Setup

The proposed solution is implemented with below specification and configuration.

- Processor: Intel i5
- Speed: 2GHz
- Memory: 8GB RAM
- Programming language: Python
- Environment: Jupyter Notebook

Libraries Used

Pandas:

It's a Python-based machine learning library. Pandas is a free and open-source programming language. Pandas is a programming language that is commonly used for dataset loading and data analytics. Pandas is used for machine learning in a variety of domains, including economics, finance, and others. It is extremely user-friendly and can display datasets in a tabular style for easier comprehension.

Sklearn:

Sklearn is one of the most essential Python libraries for machine learning. Sklearn includes several tools for statistical classification, modelling, regression, dimensionality reduction and clustering.

Numpy:

Numpy is a Python-based machine learning package. In Python, Numpy is used to deal with arrays. NumPy is used for all calculations using 1-d or 2-d arrays. Numpy also has routines for working with linear algebra and the Fourier transform.

MAPTlotlib:

MAPTlotlib is a library for data visualization. It's a Python open-source module for plotting graphs from model results. These diagrams can aid in comprehending the circumstance of the outcomes. For easier comprehension, several components of the results can be graphically formatted.

Evaluation

In this section, we use different models of machine learning for evaluating the accuracy. It has been explained about the different models in below sections. Where in this project the models are examined, with accuracy as the primary metric. In final stage we have compared the model accuracy. In all circumstances the testing and training datasets are splinted into 20:80 ratio.

Experiment 1/ Feature Distribution

Here in below figure shows how the data is distributed and how features are related to one another, a few plots and graphs are given.

Experiment 2/ Decision Tree Classifier

The method runs through all potential tests to discover the one that is most informative about the target variable to build a tree. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 82.6% and 81%. Below is the execution of Decision tree classifier algorithm. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model.

Experiment 3/ Random Forest Classifier

We can limit the amount of over fitting by averaging the outcomes of numerous trees that all operate well and over fit in diverse ways. To construct a random forest model, you must first determine the number of trees to construct. They are incredibly powerful, frequently operate effectively without a lot of parameters adjusting, and don't require data scalability. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 83.4% and 81.4%.

Experiment 4/ MLP

MLPs can be thought of as generalized linear models that go through numerous phases of processing before deciding. Below is the execution of MLP algorithm. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 86.3% and 85.9%.

Experiment 5/ XGBoost

Below is the execution of XGBoost algorithm. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 86.4% and 86.6%.

Experiment 6/ Auto encoder

The auto-encoder must learn to encode the input to the hidden neurons with fewer neurons. In an auto encoder, the predictors (x) and output (y) are identical. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train,

X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 81.8% and 81.9%.

Experiment 7/ SVM

An SVM training algorithm creates a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples that are individually designated as belonging to one of two categories. To generate model various parameters are set and the model is fitted in the tree. The samples are divided into X and Y train, X and Y test to check the accuracy of the model. Where we are predicting the accuracy of the model on the samples collected on both trained and test samples. On this we found accuracy of test and training datasets are 81.8% and 79.8%.

Result and Discussion

As a final step of evaluation, we have compared all the machine learning models. A data frame is constructed to compare the models' performance. The lists constructed to store the model's findings are the columns of this data frame. Below is the code snippet for comparing the models an accuracy result. The accuracy of the training and test datasets by the individual models.From our project we came to know that XGBoost ML model has the high accuracy compared to other model and the least accuracy is SVM. The XGBoost technique has 17the highest values in all the performance metrics used, indicating that it is the most robust of the complete algorithm, according to the experimental data. This could be due to the strategy used by the proposed model to avoid over fitting. Knowing that one of the most common problems with SVM, MLP, and Random forests is that they over fit for some datasets with poor classification objectives. The XGBOOST rows sub sampling, regularization term, shrinkage parameters, and are column sub sampling all approaches that XGBOOST uses to avoid over fitting. Auto encoder has the same issue in that it requires a lot of memory to store the structure and its execution is slow, but XGBOOST provides a lot of advantages over typical gradient boosting implementations. These are the main features of XGBoost to achieve more

accuracy rate compared to other models.

Conclusion and Future Work

A comparison of machine learning techniques for URL prediction is offered in this research. The major goal is to ensure security and prevent the user from gaining access to their sensitive data. It is possible to determine whether a website is legitimate or not using machine learning algorithms. With the comparison with other models in the research we found XGboost Classifier has a high accuracy by including 16 features. This project can be expanded upon by generating browser extensions and adding a graphical user interface. Using the current model, we can classify the Supplied URL as legitimate or phishing.

2.2. A Survey of Machine Learning-Based Solutions for Phishing Website Detection

Abstract

With the development of the Internet, network security has aroused people's attention. It can be said that a secure network environment is a basis for the rapid and sound development of the Internet. Phishing is an essential class of cyber criminals which is a malicious act of tricking users into clicking on phishing links, stealing user information, and ultimately using user data to fake logging in with related accounts to steal funds. Network security is an iterative issue of attack and defense. The methods of phishing and the technology of phishing detection are constantly being updated. Traditional methods for identifying phishing links rely on blacklists and whitelists, but this cannot identify new phishing links. Therefore, we need to solve how to predict whether a newly emerging link is a phishing website and improve the accuracy of the prediction. With the maturity of machine learning technology, prediction has become a vital ability. This paper offers a state-of-the-art survey on methods for phishing website detection. It starts with the life cycle of phishing, introduces common anti-phishing methods, mainly focuses on the method of identifying phishing links, and has an in-depth understanding of machine learning-based solutions, including data collection, feature extraction, modeling, and evaluation performance. This paper provides a detailed comparison of various solutions for phishing website detection

INTRODUCTION

The Internet has become an indispensable part of people's lives. It is impossible to imagine the world without the Internet. The January 2021 global digital population report shows that there are 4.66 billion active Internet users worldwide, accounting for 59.5% of the global population. Among them, 92.6% of users use smartphones to connect to the Internet. The Internet has completely changed the way people live and work, such as information communication, shopping, chatting, and office work. Due to the pandemic that started at the end of 2019, many traditional industries have shifted from offline services to online services, such as catering and retail. Netizens have left a lot of sensitive data on the Internet, such as usernames, account names, passwords, privacy questions, personal information, and credit card information. Cyber criminals obtain this information through various illegal means and forge these users to carry out illegal activities on the Internet. In the early days of the invention of the Internet, network security issues have already appeared. With the development of the Internet, network attack techniques have also changed rapidly, which has brought many challenges to network security. According to the methods and forms of network attacks, cybersecurity issues are mainly divided into the denial-of-service attack (DoS), man-in-the-middle (MitM), SQL injection, zero-day exploit, DNS tunneling, phishing, and malware categories. Phishing is a network attack that combines social engineering and computer technology to steal the sensitive personal information of users. Attackers solicit individuals to click phishing links by sending them emails, SMS, or social media messages with deceptive content. Phishing has been around for more than 30 years, and a large number of users are deceived every year, causing economic losses. In particular, in 2020, the number of phishing attacks increased tremendously [2]. Since the COVID-19 pandemic, government departments in many countries have introduced financial assistance programs. Cybercriminals use phishing to obtain sensitive personal information, thereby fraudulently applying for government subsidies such as unemployment benefits. Among the cyber-attack complaints reported by the U.S. public in 2020, phishing network complaints accounted for the highest proportion [2]. In addition, the APWG phishing activity trends report for 2020 shows that the number of phishing attacks almost doubled in 2020 over the course of the year [3]. Anti-phishing strategies involve educating netizens and technical defense. In this paper, we mainly review the technical defense methodologies proposed in recent years. Identifying the phishing website is an efficient method in the whole process of deceiving user information. Many academic research and commercial products were published for detecting phishing websites. The traditional methods are list-based solutions that collect valid, legitimate websites to a whitelist or verified phishing websites to a blacklist and widely share the list to avoid other users being attacked. These approaches effectively prevent the reuse of the same phishing website URL, reducing the number of affected users and losses. It is widely used in

real-time defensive actions since the computational time cost is very low in a single-string match algorithm. However, these methods have a significant disadvantage: the inability to detect new phishing URLs. Therefore, some innocent users will be attacked before the link is added to a blacklist. Some researchers proposed rule-based methods to recognize new fake websites. This method involved security expert experience and website analysis of phishing sites. According to the W3C standard, a basic URL consists of the protocol, subdomain, domain name, port, path, query, parameters, and fragment. Primarily, rules are generated from the components of URLs, such as if the domain name is similar to other legitimate domains. In these rules, some need to request third-party services to obtain information, such as what is the registration date of the domain. When the rules are published in some technical articles, phishers learned them and then figured out new phishing URLs which did not match the rules. Afterward, cybersecurity specialists developed more rules, some based on the source codes of web pages. Along with the development of machine learning techniques, various machine learningbased methodologies have emerged for recognizing phishing websites to increase the performance of predictions. Phishing detection is a supervised classification approach that uses labeled datasets to fit models to classify data. There are various algorithms for supervised learning processes, such as naïve Nayes, neural networks, linear regression, logistic regression, decision tree, support vector machine, K-nearest neighbor, and random forest. A practical product needs a robust solution that generally should satisfy two requirements. The first is a high accuracy and low false warning rate. Improving the model's performance requires a substantial dataset, especially for neural networks with complex structures. In addition, computational time is a crucial factor for real-time detection systems. The primary purpose of this paper is to survey effective methods to prevent phishing attacks in a real-time environment. It presents the basic life cycle of a phishing attack as the entry point, focusing on the phase when a user clicks on a phishing link and using technical methods to identify the phishing link and alert the user. In addition to the commonly used blacklist matching and recognition methods, this paper provides an in-depth explanation of the machine learning-based URL detection technology. This paper presents the state-ofthe-art solutions, compares and analyzes the challenges and limitations of each solution, and provides ideas for research directions and future solutions.

The main contributions of this paper are the following:

1. A phishing life cycle to clearly capture the phishing problem;
2. A survey of major datasets and data sources for phishing detection websites;
3. A state-of-the-art survey of machine learning-based solutions for detecting phishing websites.

Background and Related Work:

Phishing is a common cyberattack performed by sending an email or a message to deceive recipients visiting a bogus page and then collecting users' sensitive data, such as usernames, passwords, and credit card numbers, for financial gain. Figure 1 demonstrates the phishing life cycle. First, an attacker creates a phishing website that looks very similar to a legitimate website. On the one hand, attackers used spelling mistakes, similar alphabetic characters, and other methods to forge the URL of the legitimate website, especially the domain name and network resource directory. For example, Although the browser on the computer can see the URL address by moving the mouse to the clickable link, it is difficult for the average user to identify these URLs with the naked eye and memory as imitating legitimate URLs. On the other hand, imitation of web content is also a key point. Typically, attackers use scripts to obtain logos, web layouts, and text from genuine web pages. Form submission pages that require user input of sensitive information are most often faked by cybercriminals, such as the login page, payment page, and find password page. *Mach. Learn. Knowl. Extr.* **2021**, *3* FOR PEER REVIEW 3 each solution, and provides ideas for research directions and future solutions. The main contributions of this paper are the following:

1. A phishing life cycle to clearly capture the phishing problem;
2. A survey of major datasets and data sources for phishing detection websites;
3.A state-of-the-art survey of machine learning-based solutions for detecting phishing websites. The rest of the paper is organized as follows. Section 2 reviews the background and related work of phishing. Section 3 lists the methodologies of detecting website phishing in terms of list-based methods, heuristic strategies, and machine learning-based solutions. In particular, the general architecture of the phishing network detecting solution based on machine learning is explained in detail. Section 4 introduces several frameworks of website phishing detection systems. Section 5 presents the state-of-the-art machine learningbased solutions, which are classified into three categories based on the number and characteristics of the learning model. Section 6 discusses the challenges of detecting phishing attacks.Perpetrators are good at building a sense of fear and urgency and gaining the user's trust via text messages. Afterward, the user clicks the link that will direct them to open a fake website. Particularly, real URL strings are hidden before redirecting to web browsers on mobile phones. The next step is collecting personal information on the fake website, which looks like the real company or organization's web page, by using a similar logo, name, user interface design, and content, commonly occurring with login,

reset password, payment, and renewal personal information. When users submit sensitive information to web servers that attackers build, criminals will receive all the data. The last step is stealing the user's account funds by using the user's real information to fake the user's request for a real website. Even some individuals are using the same usernames and passwords for multiple websites. In this way, the attacker steals multiple accounts from the user. Some phishers use stolen data for other criminal activities. Since the phishing technique was recorded in a paper in 1987 , phishing methods have changed with the development of the Internet. For example, when online payment becomes popular, attackers target online payment phishing. According to the 2020 Internet Crime Report, the Internet Crime Complaint Center (IC3) received 791,790 cyberattack complaints, of which phishing scams accounted for approximately 30%, becoming the most complained about type of cybercrime and causing more than USD 54 million in losses. Therefore, to individuals who surf on the Internet, distinguishing between real and fake web pages is vital. Users need visual tools to help users identify phishing websites.

Anti-Phishing:

As we can see from, there are five steps before an attacker steals the money from the user's account or uses the information for other attacks. Therefore, blocking any step could stop a phishing attack. Here, we discuss the method of anti-phishing starting from each step.

Web Scraping:

Although it is hard to prevent perpetrators from creating web pages, some techniques could increase their costs. Attackers will use scripts to write crawlers to obtain legal web pages' content automatically and then intercept useful information and copy it to phishing web pages. Therefore, legitimate websites could prevent web scraping by several techniques is respect to obfuscation using CSS sprites to display important data, replacing text with images.

Spam Filter:

Spam filtering techniques are used to identify unsolicited emails before the user reads or clicks the link. Some mainstream email services have integrated spam filtering components, such as Gmail, Yahoo, Outlook, and AOL. The initial filters relied on blacklists or whitelists and empirical rules. With the development of artificial intelligence technology, some filters also integrate intelligent

prediction models based on machine learning to filter out spam that is not on the list. For example, Gmail could block approximately 100 million extra spam emails daily with the machine learning-based spam filter.

Detecting Fake Websites:

When users visit a phishing web page that looks like a legitimate website, many people do not remember the legitimate website's domain name, particularly for some start-ups or unknown companies, so users cannot recognize the phishing website based on the URL. Some web browsers integrate a security component to detect phishing or malware sites, such as Chrome, which will display warning messages when one visits an unsafe web page. Google launched Google Safe Browsing in 2007, and it is integrated into many Google products, such as Gmail and Google Search. Google Safe Browsing is a security component based on a blacklist that contains malware or phishing URLs. In addition, there are several web browser extensions for detecting phishing websites. However, the blacklist or whitelist-based solutions are invalid for unknown phishing websites. Fortunately, the rapid development of artificial intelligence technology has brought new ideas and solutions to detecting phishing attacks. The predictive model based on machine learning can identify phishing links that are not on the whitelist and circumvent existing rules.

Second Authorization Verification:

After the attacker obtains the user's sensitive data, the next step is to use the data to log in to the legitimate website, operate the account, and steal funds. Therefore, when the website detects that the IP address and device information of the user who is logging in does not match the commonly used information, it becomes crucial to add steps to verify the authenticity of the user. Usually, the extra verification's are dynamic and biological, such as facial movement, expression recognition, or voiceprint recognition.

Related Work

Many survey papers have been published introducing and comparing different solutions for detecting phishing websites. Basit et al. reported a survey on artificial intelligencebased phishing detection techniques. The authors analyzed the harm and trends of phishing attacks from statistical phishing reports. They collected major communication media and target devices during

phishing attacks and listed various attack techniques. The paper focuses on anti-phishing measurements, which are classified into four sections: machine learning, deep learning, hybrid learning, and scenario-based. Each section presents several major algorithms and conducts a comparison among those algorithms. In addition, they draw several conclusions by reading various state-of-the-art solutions, stating that machine learning-based solutions are widely used and effective, the feature selection process contributes high-grade performance, high accuracy often requires more computing resources, and the random forest model obtains the highest accuracy. Singh et al. conducted a review on machine learning-based phishing detection. The authors introduced a brief history of phishing and major phishing attack reports. In the paper, phishing attacks are divided into two types: social engineering attacks and malwarebased phishing. They classified features into three categories—source code features, URL features, and image features—which are all based on rules. In 2020, Vijayalakshmi et al. presented a survey on major detection techniques and taxonomy for detecting phishing. A statistical report from APWG shows the trend of phishing attacks from 2017 to 2019. In the paper, a taxonomy of automated phishing detection solutions was introduced, which classified all the solutions into three categories depending on the input parameters: web address-based methods, webpage content-based solutions, and hybrid approaches. According to the techniques applied in the solutions, web address-based approaches were divided into list-based, heuristic rule-based, and learning-based approaches, and web content-based solutions were separated into rulebased and machine learning-based solutions. The authors listed most of the state-of-the-art methodologies for each category and interpreted the details of every solution. After comparing all methods by several evaluation metrics, such as classification performance, limitations, third-party service independence, and zero-hour attack detection, they suggested that hybrid approaches would obtain a high accuracy rate and be suitable for real-time systems and that deep learning-based solutions will be a valuable direction in the future.Kalaharsha and Mehtre surveyed phishing detection solutions that were classified into several categories based on the techniques and input parameters applied. In the paper, different types of phishing attacks and three phishing techniques are introduced. The authors listed 18 methods and 9 datasets for detecting phishing websites and compared the accuracy performance among all the models. In addition, some challenges are presented in the paper, such as reducing false-positive rates and over fitting.More recently, Jain and Gupta presented a comprehensive survey on analyzing phishing attack techniques, detection methods, and some existing challenges [12]. They imported statistical reports and motivation of phishing attacks and presented different phishing attack techniques on PCs and smartphones. Then, the authors introduced various defense methods and compared existing anti-phishing

approaches published from 2006 to 2017 for their advantages and limitations. Afterward, several major challenges were presented, such as selecting efficient features, identifying tiny URLs, and detecting smartphones.

Methodologies of Phishing Website Detection

Since phishing is a social engineering issue, effective countermeasures are built for different aspects in terms of education, legal supervision, and technical approaches. This survey focuses on technical strategies for detecting phishing websites. The methodologies of detecting phishing websites are developed, which are divided into three categories: list-based, heuristic, and machine learning methods. The list-based approaches consist of whitelists and blacklists that have been manually reported and confirmed by systems. A whitelist is a set of validated legitimate URLs or domains. Obviously, a blacklist is a group of approved phishing websites. Since one user reported and verified the website as a phishing website, the URL will be added to the blacklists, which could be used to prevent other users from being disrupted. Heuristic strategies identify a phishing web page depending on a group of features extracted from the textual contents of the website and compare the features with the legitimate one. The idea of the approach is that the attackers usually deceive users by imitating well-know websites. The machine learning methods also depend on the features from the website, build the model to learn from a batch of data with structured features, and then predict if the new website is a phishing website. In the machine learning area, detecting phishing websites is a classification problem.

List-Based Approaches:

Jain and Gupta proposed an auto-updated, whitelist-based approach to protect against phishing attacks on the client side in 2016. The experimental results demonstrate that it achieved 86.02% accuracy and less than a 1.48% false-positive rate, which indicates a false warning for phishing attacks. The other benefit of this approach is fast access time, which guarantees a real-time environment and products.

Heuristic Strategies:

Tan et al. introduced a phishing detection approach named PhishWHO, which consists of three phases. First, it obtains identity keywords by a weighted URL token system and ensembles the N-

gram model from the page's HTML. Secondly, it puts the keywords into mainstream search engines to find the legitimate website and the legal domain. Next, it compares the legal domain and the target website's domain to determine if the target website is a phishing website or not. Chiew et al. used a logo image from the website to distinguish if the website was legal. In this paper, the authors extracted a logo from web page images by some machine learning algorithms and then queried the domain via the Google search engine with a logo as a keyword. Therefore, some researchers also called this category search engine-based approach.

Machine Learning-Based Methods:

Machine learning-based countermeasures are proposed to address dynamic
phishing attacks with higher accuracy performance and lower false positive rates than other methods. Consequently, the machine learning approach consists of six components: data collection, feature extraction, model training, model testing, and predicting.The flowchart of each part. Existing machine learning-based phishing website detection solutions are based on this flowchart to optimize one or more parts to obtain better performance.

Data Collection and Feature Extraction:

Data are the source of each approach and proves to be a vital influence for the performance. There are two methods to collect data: loading published datasets and pulling URLs directly from the Internet.In these three published datasets, every row's data object contains several features extracted from a URL and a label of classes. The original URL strings could be collected from websites by running open API or data mining scripts. Mohammad et al. proposed an automatic technique to extract phishing website features and weigh the importance of each feature in 2012 [17]. In that paper, the authors collected 2500 phishing URLs from the phishTank archive [18] and extracted 17 features which were classified into three categories: address bar-based features, abnormal-based features, and HTML and JavaScript-based features. Most of the features were automatically extracted from the URL and the source code of the web page without relying on third-party services. However, the age of the domain and DNS record were extracted from the WHOIS database. The rank of the web page was obtained from the Alexa database. Meanwhile, the authors described an IF-ELSE rule and set a weight for each feature. The weight of a feature came from the calculation of the feature value for phishing accounts for the total number of phishing links. Each feature's value could be numeric as an element of the set {1, 0, −1},

respectively, each standing for legitimate, suspicious, and phishing in turn. In 2015, Mohammad et al. published a phishing website dataset on the UCI Machine Learning Repository, which became a foundation for machine learning-based phishing detection solutions and was widely used in many related research areas, containing 11,055 instances with 30 features. Furthermore,Choon published a phishing dataset on Mendeley in 2018, containing 10,000 data rows with 48 features extracted from phishTank and OpenPhish for phishing webpages and Alexa and Common Crawl for legitimate webpages, each having 5000 websites. Machine learning flowchart for detecting phishing websites. OpenPhish, and Spamhaus.org for phishing URLs and dmoztools.net, Alexa, and Common Crawl for legitimate websites, and then parsed the features by themselves. With the successful development of the natural language processing (NLP) technique, many researchers capture character-level features from URL strings based on the NLP and then feed them into deep learning models to increase the accuracy. The significant advantages of this method are irrelevant cybersecurity expertise and not relying on thirdparty network services [24]. Since the characters in the URL are continuous, it is difficult to distinguish words and have no semantics. Character-level features are used, such as character-level TF-IDF features. TF-IDF means Term Frequency–Inverse Document Frequency. The character level stands for each character as a term.

Feature Selection

Feature selection is the process of automatically selecting important features which contribute the most to the machine learning model. Having closely relevant features in the input can enhance the performance of the model, decrease training time (especially in deep learning models), and reduce overfitting issues. Generally, feature selection methodologies could be classified into three categories: the filter method, wrapper method, and embedded method. Zamir et al. utilized recursive feature elimination (RFE), information gain (IG), and relief ranking to omit redundant features for phishing detection. Furthermore, they introduced principal components analysis (PCA) for analyzing attributes [26,27]. IG is an indicator that tells us the importance of features by calculating class probability, feature probability, and class probability under a feature condition. RFE is a widely used feature reduction algorithm to remove the least essential features in the training process until the error rate meets expectations. A relief ranking filter is a feature value filtering algorithm that calculates the feature value score by comparing the feature values of two adjacent data points discovered by the nearest neighbor search algorithm and then sorts them to obtain the feature value weight according to the score. Shabudin et al. used this algorithm to apply to the UCI dataset for phishing website classification. After the feature selection process, they

obtained 22 features with weights ranking and removed 8 redundant features of zero scores [28]. Zabihimayvan and Doran applied Fuzzy Rough Set (FRS) theory to select important features from the UCI dataset and Mendeley dataset for phishing detection applications Fuzzy Rough Set (FRS) theory is an extension of Rough Set (RS) theory. RS is a method to find a decision boundary by calculating the equality of each data point based on certain features and the same classes, such as websites A and B both being phishing websites and their features a and b having the same value. RS is suitable for the original UCI dataset in which the features are utilized as a discrete value; that is, they are an element of set {−1, 0, 1}. However, after the dataset executes the nominalization process, the value of the feature is transferred to a continuous number from 0 to 1, and the FRS strategy is applied. El-Rashidy introduced a novel technique to select features for a web phishing detection model in 2021 [29]. The feature selection method contains two phases. The first phase calculates each feature's absence impact by training the random forest model with a new dataset that removes one feature and figures out the accuracy. After the absence of each element in the loop, a feature queue ranked from high to low accuracy is obtained. The second stage is to train and test the model by starting from one feature, adding a new feature from the ranked feature list each time to form a dataset, calculate the accuracy of each time, and finally find the feature subset with the highest accuracy. This method works to select the most effective feature subset. However, since each new dataset must go through the algorithm training and testing process, a high computational complexity and a long calculation time are involved. For example, if the UCI dataset has 30 eigenvalues, then the first stage loops 30 times, the second stage loops 30 times, and the tree algorithm training must be performed each time. Therefore, this methodology is suitable for small feature sizes and single classifiers.

Modeling:

Machine learning-based models can be classified into three categories: single classifier, hybrid models, and deep learning. Hybrid models combine more than one algorithm applied to the training process. Phishing website detection is a binary classification problem. Some widely used classification algorithms are listed below.

SVM:

A support vector machine (SVM) is a supervised learning algorithm that classifies data points into two sections and predicts new data points belonging to each section. It is suitable for linear binary classification, which has two classes labeled, and the classifier is a hyperplane with N dimensions relevant to the number of features. The core idea of this algorithm is to maximize the distance between the data point and the segmentation hyperplane. For example, there are two classes—phishing and legitimate—and a 29-dimension hyperplane when we use the UCI dataset for training the SVM model.

Decision Tree:

A decision tree is a popular machine learning algorithm, and the model logic is a tree structure. Each node in the decision tree is a feature; each stem presents a feature value and a possibility, and the last node presents the result. The more straightforward tree structure tends to have better performance. When trees grow very deep, it likely leads to overfitting trainingdatasets.

Random Forest:

A random forest is an ensemble of decision trees for classification and regression. Random forests reduce the overfitting problem by classifying or averaging the output of individual trees in training processing. Therefore, random forests generally have higher accuracy than decision tree algorithms.

k-NN:

A k-nearest neighbors' algorithm (k-NN) is a non-parametric classification
algorithm that makes predictions by finding similar data points through calculating the distance between the target and the nearest neighbors. There are some methods to calculate the distance with respect to the Euclidean distance for continuous data and the Hamming distance for discrete values. In particular, it does not have a training process, and each prediction will take a long time. Therefore, this algorithm is generally not suitable for real-time scenarios.

Bagging:

Bagging, also called bootstrap aggregating, is an ensemble meta-learning algorithm for improving other machine learning algorithms' performance in classification and regression. The bootstrapping procedure divides the original training dataset into N pieces and uses resampling techniques to generate the same size of the original datasetin each piece and then conducts classification in N iterations that could be executed in parallelization. Finally, the aggregating process combines N classifier outputs by averaging or voting. Naive Bayes: A naive Bayes classifier is a probabilistic statistical algorithm based on Bayes' theorem with robust independence features. Bayes' theorem is a conditional probability theory. It is also called simple Bayes and independence Bayes. In recent years, more and more researchers have used hybrid classification in phishing website detection approaches to achieve higher performance and lower computational times than single classifiers. Most hybrid models are based on a primary learner, with the addition of an algorithm for feature selection or optimizing the initialization parameters of the basic algorithm, such as hyperparameters for neural networks. Since the rapid development of deep learning and the success of natural language processing (NLP), researchers have proposed diverse deep learning models which derive information and sequential patterns of URL strings without depending on the source code features extracted from the web page content. It does not require professional cybersecurity knowledge of phishing and depends on third-party services to capture characteristics. Some broadly used deep learning algorithms are listed below.

CNN:

A convolutional neural network (CNN) is a feed forward deep learning algorithm and is widely used in image classification. The regular architecture of a CNN consists of multiple layers, followed by the input layer, hidden layers, and output layer. Commonly, the hidden layers have convolutional layers, pooling layers, and fully connected layers.

RNN:

A recurrent neural network (RNN) is a deep neural network with an internal memory function to handle diverse length sequences of inputs, such as text. Therefore, it has been successfully applied in text mining. summary of these algorithms based on the same dataset. We used the Big O notation to measure the computational complexities of machine learning algorithms. The complexity of a deep neural network depends on the architecture of the networks. Generally, it

needs to compute the activation function of all neurons. Interpretability presents the difficulty of understanding how the model works. Traditional machine learning algorithms are user-friendly models. In deep neural networks, it is hard to know which neuron is playing what role and which input feature contributes to the model output. In contrast, deep neural networks require more training data than other algorithms to obtain acceptable performance. The significant advantage of deep neural networks is dealing with text data, such as URL strings.

Performance Evaluation:

The evaluation of performance was carried out during the testing process. The original dataset would be divided into training data and test data, usually 80% and 20%, respectively. When evaluating the classifier's behavior on the testing dataset, there were four statistical numbers: the number of correctly identified positive data points (TP), the number of correctly identified negative data points (TN), the number of negative data points labeled by the classifier as positive (FP), and the number of positive data points labeled by the model as negative (FN). There are several broadly used metrics to evaluate performance. The classification accuracy is the ratio of correct predictions to total predictions:

accuracy = TP + TN /TP + TN + FN + FP

In binary classification cases, it is known that random selection has 50% accuracy. In unbalanced datasets, sometimes high accuracy does not mean that the model is excellent. For instance, among the 10,000 data, 9000 were legitimate websites, and 1000 were phishing websites, so when the prediction model did nothing, it could reach 90%. Accuracy is misleading when the class sizes are substantially different. Precision is the percentage of correctly identified positive data points among those predicted as positive by the model. The number of false-positive cases (FP) reflects the false warning rate. In real-time phishing detection systems, this directly affects the user experience and trustworthiness:

Precision = TP/TP + FP

The recall is the portion of positive data points labeled as such by the model among all truly positive data points. The number of false-negative cases (FN) represents the number of phishing URLs that has not been detected. Leak alarms mean that users are likely to receive an attack that could result in the theft of sensitive information. Misleading users can do more harm to users than not detecting them:

Recall = TP /TP + FN

The F-measure or F-score is the combination of precision and recall. Generally, it is formulated as shown below:

$F\beta = (\beta 2 + 1) \times Precision \times Recall / \beta 2 \times Precision + Recall$ $\beta \in (0, \infty)$

Here, β quantifies the relative importance of the precision and recall such that

$\beta = 1$

stands for the precision and recall being equally important, which is also called F1. TheF-score does the best job of any single statistic, but all four work together to describe the performance of a classifier:

$F1 = 2 \times Precision \times recall / Precision + recall = TP / TP + 1/2 (FP + FN)$

In addition, many researchers use the N-fold cross-validation technique to measure performance for phishing detection [4,30,31]. The N-fold cross-validation technique is widely used on small datasets for evaluating machine learning models' performance. It is a resampling procedure that divides the original data samples into N pieces after shufflfling the dataset randomly. One of the pieces is used in the testing process, and others are applied to the training process. Commonly, N is set as 10 or 5.

Frameworks of Phishing Website Detection Systems:

The goal of anti-phishing research is to prevent individual Internet users from suffering phishing attacks. With the development of anti-phishing research, phishing attackers are constantly updating their technology. The naked eye does not recognize many phishing links well, and individual netizens need tools to help identify them. Due to the tools and methods of the phishing network, many researchers naturally think of expanding on the browser. The following two methods are based on the browser.

Anti-Phishing Web Browser:

In 2020, HR et al. built a web browser with a phishing detection component [32]. The regular web browser had two core engines—a browser engine and a render engine—which are responsible for connecting to the Internet to fetch the web page via the URL, parsing the web page by XML, HTML, CSS, JAVASCRIPT interpreters, storing cookies, etc. The proposed browser added an intelligent engine to detect phishing websites between the browser engine and render engine. When a user

input a URL, the intelligent engine started to predict if the target website was a phishing website and afterward sent the result to the render engine. If the predicted result showed a phishing website, the render engine would pop a warning message to the user interface. This paper used the random forest algorithm to train the model, and it obtained 99.36% accuracy and a 0.64% false-positive rate on the UCI dataset with 30 rule-based features.

Web Browser Extensions :

Armano et al. introduced a real-time client-side phishing prevention solution. The approach contains a built-in JavaScript front end and a built-in Python back end. The front end collects the web page source code and handles the user interface and interaction with the back end, analyzing the website and predicting if it is a phishing website. The backend consists of a disputer for checking against the whitelist, a phishing detector for predicting the website's legitimacy, and a target identifier to find the legitimate website relevant to the input URL based on the logo, keywords, and other content. The phishing detector is implemented by an existing solution that uses the gradient boosting algorithm as the classifier [34]. The authors experimented with 200 phishing websites to monitor the response time. The results showed that the response time for a phishing URL was longer than a legitimate one, which was approximately 2 s, and the appearance of the alert cost occurred in less than 500 ms. With the rapid development of the mobile Internet, many user behaviors have shifted from the PC to the smartphone. Therefore, phishing website monitoring on mobile phones is vital.

Mobile Applications:

Kadhim et al. developed a web browser application on Android smartphones to predict phishing websites based on the UCI dataset with 30 features extracted from the URL and source code. The application compared different classification algorithms in training processing, such as the decision table, J48, SVM, Bayes Net, and random forest model, which outperformed the others with 97% accuracy. The authors conducted the experiments on Samsung and Nexus phones running the Android 5.1 operating system. In addition to the framework mentioned in academic papers, there are also several published Internet products.

State-of-the-Art Machine Learning-Based Solutions:

In recent years, massive phishing detection solutions were proposed and achieved high accuracy. It is believed that the recently proposed methods are more advanced. Several major state-of-the-art methodologies are listed below, and they are classified into three categories.

Single Classifier:

In 2021, Gupta et al. developed a lightweight phishing detection approach and achieved 99.57% accuracy with the random forest algorithm. The authors extracted 19,964 instances with 9 lexical features from the ISCX-URL-2016 dataset published by the University of Canada Brunswick. The ISCX-URL-2016 dataset contains more than 35,300 legitimate URLs and approximately 10,000 phishing URLs taken from an active repository of phishing sites .To balance the distribution of the two classes, the authors randomly filtered 10,000 benign URLs and 9964 phishing URLs. Further more, the Spearman correlation algorithm and K best algorithm are applied to figure out the feature importance. Based on other previous research, nine lexical features from URLs were proposed in the paper. Afterward, they cleaned the data by replacing the null and unlimited values with mean values and normalized them by scaling the values between 0 and 1. Normalization is one of the important data pre processing procedures to guarantee that one feature is not dominated by others. In addition, they used a one-hot encoding algorithm to transfer the labels to numerical values. Once the dataset is regularized, it is divided into a training dataset and a testing dataset with eight-to-two ratios. In the process of modeling, they compared four single classifiers with the performance and computational time. Finally, it was concluded that random forest had the highest accuracy rate and the lowest false positive rate. However, in terms of response time, SVM performed better.

Hybrid Methods

In 2020, Alsariera et al. proposed four hybrid models named ABET, RoFET, BET, and LBET, each combining a meta-learner model and the extra tree algorithm, which is the basic classifier. Four meta-learner models, called Adaboost.M1, Rotation Forest, Bagging, and LogitBoost, were implemented by a meta-algorithm or metaheuristic, a high-level procedure designed to find an optimal solution for an optimization problem. This paper used 10-fold cross-validation to resample

the UCI dataset and then iterated 10 times for training and testing the extra tree model, evaluated based on a weighted average value. The Adaboost.M1 model was used with a base classifier to improve performance by iterating 100 times to adjust the weights. The RoFET model used a principal component filter in the training process to achieve a high true-positive rate and decrease bias. The BET combined the bagging algorithm and extra tree algorithm executed 150 times over a resampled dataset. The LBET is a logistic regression extra tree that conquers abnormal data points, such as noise and outliers. The experimental results demonstrated that all four fusion models obtained significant performance, with over 97% accuracy, false-negative rates less than 0.038, and false-positive rates less than 0.019. Zamir et al. introduced diverse machine learning algorithms for detecting phishing websites, comparing accuracy performance from the single classifier to the stacking models. First, the authors conducted a data preprocessing procedure containing feature selection, nominalization, and principal components analysis (PCA), a dimension reduction method. The feature selection process involved a variety of algorithms in analyzing the importance of features based on the UCI dataset, such as IG, GR, Relief-F, and RFE. In comparing the experimental results, they concluded that RFE was the efficient algorithm to eliminate unimportant features. Afterward, the features were used to fit the stacking model with a 10-fold cross-validation technique. They built two stacking models; one was combined random forest (RF), neural network (NN), and bagging (Bagging) algorithms, and the other was associated with the k-nearest neighbors, random forest, and bagging algorithms. The RF-NN-Bagging approach outperformed all other models introduced in the paper with respect to accuracy performance, which was 97.4% .adapted from depicts the proposed framework.

Deep Learning:

Deep learning is a subset of machine learning which is built with deep structured
architectures. There are some commonly used deep learning algorithms, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and long short-term memory (LSTM) networks. With the rapid development of natural language processing (NLP) and deep learning algorithms, various deep learning-based solutions are introduced for phishing detection. hows the basic architecture of deep learning-based approaches. Deep learning for phishing detection. Ali and Ahmed developed an intelligence phishing detection model which combined deep neural networks (DNNs) and genetic algorithms (GAs). A DNN is a well-known deep learning technique with more than two hidden layers, an input layer, and an output layer, commonly used to classify multiple labels from big data. The GAs are inspired by the biological evolution of the genes in

nature and are widely used for optimization problems that aim to minimize or maximize the value of objective functions under some restraints. In this approach, the authors regarded the problem of feature selection as an optimization problem. Mathematically speaking, the objective function minimizes the number of features, and the constraint function is the accuracy of the classification model. Meeting performance requirements with minimal features reduces the model training time and could remove the noisy data. Therefore, the GA was applied to find the optimal subset of features by computing the accuracy of the DNN model in each generation. A chromosome represents a group of features, and each gene with a binary value stands for each feature, where one is for selecting this feature and zero is not. The classification phase used the selected features as input features and the UCI dataset as a training dataset to fit the DNN model. However, the GA-DNN model got a relatively low accuracy result, which was 89%. It is known that hyperparameters and the size of a training dataset significantly affect the performance of deep learning models. In 2020, Aljofey et al. proposed an efficient convolutional neural network (CNN) model for phishing detection only based on URLs. They extracted character-level features from the original URLs, which were collected from different phishing websites and benign websites.The experimental results showed that this model obtained an accuracy of 95.02% on their own dataset with 318,642 instances. Wang et al. introduced a fast model called PDRCNN that used the URL string as an input, extracted features by an RNN and CNN, and then classified them with the Sigmoid function. The authors collected approximately 500,000 instances from Alexa.com and phishTank.com and extracted semantic features based on the word embedding technique, encoding the URL string to a tensor, an input of the RNN model. A bidirectional LSTM network algorithm implemented the RNN architecture to extract global features, which were the inputs of the convolutional neural network. The final one-dimensional tensor represented a group of features generated through multiple convolutional and max-pooling layers. Finally, the one-dimensional tensor was fed into a fully connected layer with a sigmoid function to classify the original input URL into the fake and phishing website. The experimental results illustrated that they achieved 95.97% accuracy.the comparison of major state-of-the-art solutions. The random forest algorithm obtained higher accuracy than other models, although it varied across datasets. The UCI dataset is widely used in different machine learning models, being friendly to novices and researchers without security experience. However, it requires a process of extracting features from a URL when it is applied to real-time systems. The feature extraction process is based on security experience rules and might depend on third-party services. Many researchers proposed fusion models which combined some feature selection algorithms and a normal classifier to enhance performance and reduce the search dimensions. In terms of accuracy, deep learning-based solutions attained low performance.

However, the significant advantage is that it is close to a real-time prediction system. Due to the model's input being the original URL string, it is independent of cyber security experience and third-party services. Once the model training is complete, the response time predicted online will be faster than traditional systems that rely on regular features.

Opportunities and Challenges :

Anti-phishing techniques have been developed for decades and are improved constantly. However, there are still several challenges or limitations of phishing website detection solutions.

High-Quality Dataset:

Effective phishing detection solutions should combine new data constantly for recognizing fresh rules and training machine learning models. Phishing and anti-phishing are always in the process of confronting each other. Attackers will adjust the generation of phishing links according to the published anti-phishing rules and methods. Likewise, anti-phishing needs to optimize models and algorithms based on new phishing data. Furthermore, the performance of machine learning-based solutions highly depends on the quality of the training dataset in terms of size and validation. The published datasets are small datasets that do not satisfy the demands of deep learning approaches. According to the power law, deep learning performance keeps rising with the increase of the training data size. Therefore, pulling phishing URLs and legitimate URLs from websites is recommended. However, this depends on the stability of the third-party services or websites.

Efficient Features Extraction and Selection:

According to published rules, it is not difficult to extract features from a URL. However, some rules depend on third-party services. Therefore, it might cost time and face unstable issues. Furthermore, it is important to calculate the weights of the features, decrease the dimensions, and reduce overfitting, which occurs in training processing.Choosing the most efficient features is a matter that requires multiple computing resources, and for different models, the weighting of features may need to be adjusted.

Tiny URL Detection:

Since tiny URLs do not present the real domain, resource direction, or search parameters, rule-based feature selection techniques might be useless for tiny URLs. Due to tiny URLs generated by different services, it is hard to convert them to original URLs. Furthermore, tiny URLs are short strings that are unfriendly for natural language processing to extract character-level features. If tiny URLs are not specially processed during data cleansing and preprocessing, they are likely to cause false or missed alarms. Internet products are also essential in terms of user experience, and users are also sensitive to false alarms of Internet security products. 6.4. Response Time for Real-Time Systems Rule-based models depend on rule parsing and third-party services from a URL string. Therefore, they demand a relatively long response time in a real-time prediction system that accepts a single URL string as an input in each request from a client. Phishing attacks spread to various communication media and target devices, such as personal computers and other smart devices. It is a big challenge for developers to cover all devices with one solution. Language independence and running environment independence should be taken into consideration to reduce system development complexity and late maintenance costs.

Conclusions and Future Work

This survey introduced the life cycle of phishing to clarify the important steps for anti-phishing. This paper focuses on the technical methodologies, particularly machine learning-based solutions for phishing website detection. Furthermore, the architecture of machine learning-based resolution shows the general components in the system. The details of each part inspire the development of high-performance phishing detection techniques. We reviewed diverse academic articles and sorted diverse data sources. It is easy to start with published datasets that are standardized based on rules generated by security experts' experience. However, these datasets contain limited instances. Small datasets affect model performance in the training process, particularly for complex structured models such as multi-layer neural networks. In addition, they are relatively old, being collected approximately five years ago. The alternative method is to collect URLs from websites that contain various verified phishing URLs, such as phishtank.com. The shortcoming is that this needs an extra feature extraction process based on rules, and it depends on some third-party services. In recent years, deep learning and natural language processing techniques have developed rapidly. Some researchers saw a URL as text information and used the NLP technique to extract character-level or word-level features to feed deep learning models for predicting phishing websites. The advantage of this solution is the independence of third-party services and needless specialist experience. The disadvantage is that the learning process will cost more time.

Anti-phishing has been around for decades, and many efficient solutions have been proposed. However, attack techniques are constantly changing, and no solution is once and for all. Our continuous research of phishing website detection to defend against phishing attacks and prevent financial losses is worth it. Researchers and security experts have contributed a lot of successful resolutions, from list-based methods and rule-based strategies to machine learning-based approaches. Various machine learning-based solutions achieved higher than 95% accuracy, which is a significant advancement. However, it is believed that the accuracy performance still has space for improvement. In addition, phishing detection is sensitive to false warnings. Furthermore, a real-time system requires very low computational time. Therefore, a robust and efficient phishing website detection system still has its challenges.

2.3. Detection of Phishing Websites using MachineLearning

Abstract

Phishing is a common attack on credulous people by making them to disclose their unique information using counterfeit websites. The objective of phishing website URLs is to purloin the personal information like user name, passwords and online banking transactions. Phishers use the websites which are visually and semantically similar to those real websites. As technology continues to grow, phishing techniques started to progress rapidly and this needs to be prevented by using anti-phishing mechanisms to detect phishing. Machine learning is a powerful tool used to strive against phishing attacks. This paper surveys the features used for detection and detection techniques using machine learning. Phishing is popular among attackers, since it is easier to trick someone into clicking a malicious link which seems legitimate than trying to break through a computer's defense systems. The malicious links within the body of the message are designed to make it appear that they go to the spoofed organization using that organization's logos and other legitimate contents. Here, we explain phishing domain (or Fraudulent Domain) characteristics, the features that distinguish them from legitimate domains, why it is important to detect these domains, and how they can be detected using machine learning and natural language processing techniques.

Introduction

Nowadays Phishing becomes a main area of concern for security researchers because it is not difficult to create the fake website which looks so close to legitimate website.

Experts can identify fake websites but not all the users can identify the fake website and such users become the victim of phishing attack. Main aim of the attacker is to steal banks account credentials. Phishing attacks are becoming successful because lack of user awareness. Since phishing attack exploits the weaknesses found in users, it is very difficult to mitigate them but it is very important to enhance phishing

detection techniques. Phishing may be a style of broad extortion that happens once a pernicious web site act sort of a real one memory that the last word objective to accumulate unstable info, as an example, passwords, account focal points,or MasterCard numbers. all the same, the means that there square measure some of contrary to phishing programming and techniques for recognizing potential phishing tries in messages and characteristic phishing substance on locales, phishes think about new and crossbreed procedures to bypass the open programming and frameworks. Phishing may be a fraud framework that uses a mixture of social designing what is additional, advancement to sensitive and personal data, as an example, passwords associate degree open-end credit unpretentious elements by presumptuous the highlights of a reliable individual or business in electronic correspondence. Phishing makes use of parody messages that square measure created to seem substantial and instructed to start out from true blue sources like money connected institutions, online business goals, etc, to draw in customers to go to phony destinations through joins gave within the phishing websites.

PROJECT DESCRIPTION

We have developed our project using a website as a platform for all the users. This is an interactive and responsive website that will be used to detect whether a website is legitimate or phishing. This website is made using different web designing languages which include HTML, CSS, Javascript and Django. The basic structure of the website is made with the help of HTML. CSS is used to add effects to the website and make it more attractive and user-friendly. It must be noted that the website is created for all users, hence it must be easy to operate with and no user should face any difficulty while making its use. Every naïve person must be able to use this website and avail maximum benefits from it. The website shows information regarding the services provided by us. It also contains information regarding ill practices occurring in today’s technological world. The website is created with an opinion such that people are not only able to distinguish between legitimate and fraudulent website, but also become aware of the malpractices occurring in current

world. They can stay away from the people trying to exploit one's personal information, like email address, password, debit card numbers, credit card details, CVV, bank account numbers, and the list goes on. The dataset consists of different features that are to be taken into consideration while determining a website URL as legitimate or phishing.

The components for detection and classification of phishing websites are as follows:

1.Address Bar based Features

2.Abnormal Based Features

3.HTML and JavaScript Based Features

4.Domain Based Features

1.Address Bar based Features

1.Using the IP address If IP address is used instead of domain name in the URL
e.g. 125.98.3.123 the user can almost be sure someone is trying to steal his personal information.

2. Long URL to hide the Suspicious Part Phishers can use long URL to hide the doubtful part in the address bar.

3. Using URL shortening services TinyURL URL shortening is a method on the World Wide Web in which a URL may be made considerably smaller in length and still lead to the required webpage.

4. URLs having @ symbol Using @ symbol in the URL leads the browser to ignore
everything preceding the @ symbol and the real address often follows the @ symbol.

5. Redirecting using // The existence of // within the URL path means that the
user will be redirected to another website.

6. Adding Prefix or Suffix Separated by (-) to the Domain The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-)
to the domain name so that users feel that they are dealing with a legitimate webpage.

7. Sub Domain and Multi Sub Domains Let us assume we have the following link:
http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD).

8. HTTPs (Hyper Text Transfer Protocol with Secure Sockets Layer) The existence of HTTPS is very important in giving the impression of website legitimacy, but this is clearly not enough.

9. Domain Registration Length Based on the fact that a phishing website lives for a
short period of time, we believe that trustworthy domains are regularly paid for several years in advance. In our dataset, we find that the longest fraudulent
domains have been used for one year only.

10.Favicon

A favicon is a graphic image (icon) associated with a specific webpage.

11.Using Non-Standard Port This feature is useful in validating if a particular service is up or down on a specific server.

12.The existence of HTTPS Token in the Domain Part of the URL The phishers may add the HTTPS token to the domain part of a URL in order to trick users.

2.Abnormal Based Features

1.Request URL

Request URL examines whether the external objects contained within a webpage such as images, videos and sounds are loaded from another domain.

2. URL of Anchor

An anchor is an element defined by the <a> tag. This feature is treated exactly as Request URL.

3. Links in <meta>, <Script> and <Link> tags Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta>tags to offer metadata about the HTML document; <Script> tags to create a client side script; and <Link> tags to retrieve other web resources. It is expected that these tags are linked to the same domain of the webpage.

4. Server From Handler(SFH)

SFHs that contain an empty string or about:blank are considered doubtful because an action should be taken upon the submitted information.

5. Submitting Information to Email

Web form allows a user to submit his personal information that is directed to a server for processing. A phisher might redirect the users information to his personal email.

6. Abnormal URL

This feature can be extracted from WHOIS database. For a legitimate website, identity is typically part of its URL.

3.HTML and JavaScript Based Features

1.Website Forwarding

The fine line that distinguishes phishing websites from legitimate ones is how many times a website has been redirected. Status Bar Customization

2. 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 on Mouse Over to hide the Link.

3. Using Pop-Up Window

It is unusual to find a legitimate website asking users to submit their personal information through a pop-up window.

4. IFrame Redirection

IFrame is an HTML tag used to display an additional webpage into one that is currently shown.

4.Domain Based Features

1.Age of Domain

This feature can be extracted from WHOIS database. Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age
of the legitimate domain is 6 months.

2. DNS Record

For phishing websites, either the claimed identity is not recognized by the WHOIS database or no records founded for the host name. If the DNS record is empty
or not found then the website is classified as Phishing, otherwise it is classified as Legitimate.

3. Website Traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit.

4. Page Rank

Page Rank is a value ranging from 0 to 1. Page Rank aims to measure how important a webpage is on the Internet.

5. Google Index

This feature examines whether a website is in Googles index or not. When a site is indexed by Google, it is displayed on search results.

6. Number of Links Pointing to Page

The number of links pointing to the webpage indicates its legitimacy level, even if some links are of the same domain.

7. Statistical-Reports Based Feature

URL parts and features

ALGORITHMS USED

Two algorithms have been implemented to check whether a URL is legitimate or fraudulent. Random forest algorithm creates the forest with number of decision trees. High number of tree gives high detection accuracy. Creation of trees is based on bootstrap method. In bootstrap method features and samples of dataset are randomly selected with replacement to construct single tree. Among randomly selected features, random forest algorithm will choose best splitter for classification. Decision tree begins its work by choosing best splitter from the available attributes for classification which is considered as a root of the tree. Algorithm continues to build tree until it finds the leaf node. Decision tree creates training model which is used to predict target value or class in tree representation each internal node of the tree belongs to attribute and each leaf node of the tree belongs to class label.

PROJECT REQUIREMENTS

Hardware Requirements:-

- 2GB RAM (minimum)
- 100GB HDD (minimum)
- Intel 1.66 GHz Processor Pentium 4 (minimum)
- Internet Connectivity

Software Requirements:-

- WINDOWS 7 or higher
- Python 3.6.0 or higher
- Visual Studio Code
- Django
- HTML
- Dataset of Phishing Websites

WORKING

• We have collected unstructured data of URLs from Phishtank website, Kaggle website and Alexa website, etc.

• In pre-processing, feature generation is done where nine features are generated from unstructured data. These features are length of an URL, URL has HTTP, URL has suspicious character, prefix/suffix, number of dots, number of slashes, URL has phishing term, length of subdomain, URL contains IP address.

• After this, an organized dataset is made in which each detail incorporates the paired (0,1) which is then passed to the various classifiers.

• Next, we train the three unique classifiers and analyse their presentation based on exactness two classifiers utilized are Decision Tree and Random Forest algorithm.

• At that point, the classifier identifies the given URL dependent on the preparation information that is if the site is phishing it prompts the user that the website is phished and if genuine, it prompts the user that the website is legitimate.

• We look at the exactness of various classifiers and discovered Random Forest as the best classifiers which gives the most extreme precision.

However, if the URL entered by a user is found to be a phishing website, a small pop-up will appear on the screen to warn the user regarding this malicious website. There are times when a user needs to access some data on that website, so he/she can select a ‘CONFIRM’ option to open the website, otherwise he/she will be sent back to the above webpage.

RESULTS:

Scikit-learn tool has been used to import Machine learning algorithms. Each classifier is trained using training set and testing set is used to evaluate performance of classifiers. Performance of classifiers has been evaluated by calculating classifier's accuracy score.

CONCLUSION:

Thus to summarize, we have seen how phishing is a huge threat to the security and safety of the web and how phishing detection is an important problem domain. We have reviewed some of the traditional approaches to phishing detection; namely blacklist and heuristic evaluation methods,

and their drawbacks. We have tested two machine learning algorithms on the 'Phishing Websites Dataset' and reviewed their results. We then selected the best algorithm based on it's performance and built a Chrome extension for detecting phishing web pages. The extension allows easy deployment of our phishing detection model to end users. We have detected phishing websites using Random Forest algorithm with and accuracy of 97.31%. For future enhancements, we intend to build the phishing detection system as a scalable web service which will incorporate online learning so that new phishing attack patterns can easily be learned and improve the accuracy of our models with better feature extraction.

FUTURE SCOPE:

Although the use of URL lexical features alone has been shown to result in high accuracy (~97%), phishers have learned how to make predicting a URL destination difficult by carefully manipulating the URL to evade detection. Therefore, combining these features with others, such as host, is the most effective approach . For future enhancements, we intend to build the phishing detection system as a scalable web service which will incorporate online learning so that new phishing attack patterns can easily be learned and improve the accuracy of our models with better feature extraction.

2.4 Detecting Phishing Websites Using Machine Learning

Abstract :

The goal of our project is to implement a machine learning solution to the problem of detecting phishing and malicious web links. The end result of our project will be a software product which uses machine learning algorithm to detect malicious URLs. Phishing is the technique of extracting user credentials and sensitive data from users by masquerading as a genuine website. In phishing, the user is provided with a mirror website which is identical to the legitimate one but with malicious code to extract and send user credentials top phishers. Phishing attacks can lead to huge financial losses for customers of banking and financial services. The traditional approach to phishing detection has been to either to use a blacklist of known phishing links or heuristically evaluate the attributes in a suspected phishing page to detect the presence of malicious codes. The heuristic function relies on trial and error to define the threshold which is used to classify malicious links from benign ones. The drawback to this approach is poor accuracy and low adaptability to new phishing links. We plan to use machine learning to overcome these drawbacks by implementing some classification algorithms and comparing the performance of these algorithms on our dataset.

We will test algorithms such as Logistic Regression, SVM, Decision Trees and Neural Networks on a dataset of phishing links from UCI Machine Learning repository and pick the best model to develop a browser plugin, which can be published as a chrome
extension.

Introduction

Financial services such as banking are now easily available over the Internet making the lives of people easy. Thus it is very important that the security and safety of such services are maintained. One of the biggest threats to web security is Phishing. Phishing is the technique of extracting user credentials by masquerading as a genuine website or service over the web. There are various types of phishing attacks such as Spear phishing, which targets specific individuals or companies, Clone phishing is a type of phishing where an original mail with an attachment or link is copied into a new mail with a different (possibly malicious) attachment or link, Whaling, etc. Phishing can lead to huge financial losses. For example, the Microsoft Consumer
Safer Index (MCSI) report for 2014 has estimated the annual worldwide impact of Phishing and other identity thefts to be nearly USD 5 Billion [1]. Similarly, the IRS has warned of a surge in phishing attacks with over 400% increase in reported cases [2]. Several solutions have been proposed to combat phishing ranging from educating the web users to stronger phishing detection techniques. The conventional approach to phishing detection has not been successful because of the diverse and evolving nature of phishing attacks. For instance, in January 2007, the total number of unique phishing reports submitted to the Anti-Phishing Working Group (APWG) was 29,930. Compared to the previous peak in June 2006, the number of submitted reports increased by 5% [3]. This happened despite taking preventive measure to thwart phishing. Upon investigation, it was found that each phishing attack was different from the other one. Thus, it becomes imperative to find a way to adapt our phishing detection techniques as and when new attack patterns are uncovered.Machine learning algorithms, which make a system learn new patterns from data, are an ideal solution to the problem of phishing detection. Although there have been many papers in recent years which have attempted to detect phishing attacks using machine learning, we intend to go one first step further and build a software tool which can be easily deployed in end user systems to detect phishing attacks.For our project, we will experiment with three machine learning algorithms on a dataset of features that represent attributes commonly associated with phishing pages, choose the best model based on their performance and build a web browser plugin which will eventually be deployed to end users. The project report has been

designed as follows; the Previous Work section describes the traditional approaches to phishing detection and some of the machine learning approaches attempted in recent years, the Proposed Approach section describes in detail our approach and what will be the end product of our project, the Dataset section describes the dataset that we are using for our project along with a list of features which will be used in our project, Machine Learning Algorithms section explains the different algorithms which we have tested with our dataset with their descriptions, the Chrome Plugin Implementation section describes the architecture of our phishing detection system and gives descriptions of the various software modules in the system, the Results section gives the results of our experiments with the algorithms with graphs plotting a comparison between the three algorithms on factors such as accuracy, sensitivity and false positive rate, and the Conclusion section summarizes our project with an outlook on future enhancements.

Proposed approach:

We propose to use machine learning to overcome the drawbacks associated with the traditional approaches to phishing detection. The problem of phishing detection is an ideal candidate for the application of machine learning solutions because of the easy availability of sufficient amounts of data on phishing attack patterns. The basic idea is to use machine learning algorithms on available dataset of phishing pages to generate a model which can be used to make classifications in real time if a given web page is a phishing page or a legitimate webpage. We intend to productionize the learned model into a software tool which can be deployed easily to end users for combating phishing attempts. For this purpose, we have chosen to implement a machine learning algorithm from scratch using JavaScript and build a Chrome extension with it. A Chrome extension will enable us to deploy the learned model easily on the Chrome Web Store, from where anyone can download and use our product for phishing detection.In order to successfully implement this project, we need to consider three constraints when choosing the machine learning algorithm for our product. First, the accuracy of the trained model should be high, as a product being used by end users in the real world should not give wrong results. Second, the algorithm which is being implemented should be able to make classifications in real-time; i.e. have very low execution time and also use less computational resources. Third, false positives and false negatives are important considerations when choosing a machine learning algorithm for the problem of phishing detection. This is because a user should not be wrongly led to believe that a phishing website is legitimate. Thus, we should look at these three constraints when selecting our phishing detection classifier.

DataSet

To evaluate our machine learning techniques, we have used the ‘Phishing Websites Dataset’ from UCI Machine learning repository. It consists of 11,055 URLs (instances) with 6157 phishing instances and 4898 legitimate instances. Each instance contains 30 features. Each feature is associated with a rule. If the rule satisfies, it is termed as phishing. If the rule doesn’t satisfy then it is termed as legitimate. The features take three discrete values. ‘1’ if the rule is satisfied, ‘0’ if the rule is partially satisfied, ‘-1’ if the rule is not satisfied.

The training dataset for our project is taken from the "Phishing Websites Data Set" of the UCI Machine learning repository. This dataset was compiled by [see acknowledgements]. The dataset consists of 11,055 entries with 6157 phishing instances and 4898 legitimate instances. Each instance consists of 30 features comprising of various attributes typically associated with phishing or suspicious web pages such as presence of IP address in the URL domain or presence of JavaScript code to modify the web browser address bar information. Each feature is associated with a rule. If the rule is satisfied, we take it as an indicator of phishing and legitimate otherwise. The dataset has been normalized to contain only discrete values. Each feature of each instance will contain ’1’ if the rule associated with that feature is satisfied, ’0’ if partially satisfied and’-1’ if unsatisfied. The features represented by the training dataset can be classified into four categories;

i) Address Bar based features

ii) Abnormal based features

iii) HTML and JavaScript based features

iv) Domain based features

A. Address bar based features :

1.1 Using IP address:

If the domain of the URL of the suspected web page contains IP address, then we take it as a phishing page. eg: http:125.98.3.123fake.html or http:x58.0xCC.0xCA.0x622paypal.caindex.html .

1.2 Long URL:

To hide suspicious part: It has been a common observance that phishing web pages usually have long URLs that attempt to hide malicious URL fragments from the user. We take the assumption that a web page with a long URL is necessarily
a phishing or suspicious site. In the event the assertion fails, i.e, for a legitimate web page with valid long URLs, the absence of other phishing attributes on the web page will balance the wrong assumption and correctly classify a legitimate web page as
non-phishing.

1.2 Use of URL shortening services:

A shortened URL hides the real URL behind a redirection hop. A web page that uses a URL shortening service such as Tiny URL is highly suspicious and is likely to be a phishing attempt. Therefore, we set the rule that if the URL has been shortened using a URL shortening service then it is a phishing page and legitimate otherwise.

1.3 Use of "@" symbol:

Needs verification The "@" symbol is a reserved keyword according to Web standards. So the presence of "@" in a URL is suspicious and the web page is taken as phishing and legitimate otherwise.

1.4 Redirection with "":

The presence of "//" in the URL path indicates the page will be redirected to another page. If the position of "//" in the URL is greater than seven then it is a phishing site and legitimate otherwise.

1.5 Adding prefix or suffix separated by "-" to the domain:

Phishers tend to add a prefix or suffix to the domain with "-" to give the resemblence of a geniune site. Eg: www.a-paypal.com

1.6 Sub domains and multi sub domains:

If a URL has more than three dots in the domain part then it is considered as a phishing site and legitimate otherwise.

B. Abnormal based features:

2.1 Request URL:

A legitimate site usually has external page objects such as images, animations, files, etc. be accessed by a request URL which shares the same domain as the web page URL. We classify sites which fail this rule as phishing.

2.2 URL portion of anchor tag:

We check if the domain in the URL portion of all anchor tags match the main URL of the page and if the anchor tag has only URL fragments or JavaScript functions.

2.3 Links in <meta>, <script> and <link> tags:

We check if the domain of the links in the <meta>, <script> and <link> tags matches the domain in the mail URL.

2.4 Server Form Handler (SFH):

When a form is submitted, some valid action must be taken. So if the action handler of a form is empty or "about:blank" or if the domain of the action URL is different from the domain of the main URL, then it is taken as a
phishing site.

2.5 Submitting Information to Email:

If the webpage contains a "mailto:" function then it is taken as a phishing site and legitimate otherwise.

C. HTML and Javascript based features :

3.1 Status bar customization:

Phishers can modify the status bar using JavaScript to show a legitimate URL. By analyzing the "on Mouse Over" events in the web page we can determine if such a modification has occurred.

3.2 Disabling right click option:

Phishers can disable the right click option to prevent the user from checking the source code of the page. This is verified by analyzing the source code.

3.3 Using pop-up window:

Legitimate sites rarely ask for user info on a pop-up window, whereas phishing sites generally use pop-up windows to get user info.

3.4 Iframe redirection:

Phishers also use Iframe tags with invisible borders to get user info and redirect to the original site. We analyze the source code to check if Iframe tags are used.

Machine Learning Implementation:

We have trained and tested supervised machine learning algorithms on the training dataset. The following algorithms were chosen based on their performance on classification problems. The dataset was split into training and test set in the ratio 7:3. The results of our experiment are given in the results section.

A. Random Forest:

Random forests are the classifiers that combine many tree possibilities, where each trees are depends on the values of a random vector sampled independently. then, all trees in the forest will have same allotment. To construct a tree, we assume that n is the number of training observations and p is the number of variables (features) in a training set. To determine the decision node at a tree we choose k « p as the number of variables to be selected. We select a bootstrap sample from the n observations in the training set and use the rest of the observations to estimate the error of the tree in testing phase. hence, we randomly choose 'k' variables as a decision at certain node in the tree and calculate the best split based on the k variables in the training set. Trees are always grown and never pruned compared to other tree algorithms. Random forests can handle large number of variables in a data set. Also, during the forest building process they generate an internal unbiased estimate of the generalization error. additionally, they can estimate missing data closely. A major disadvantage of random forests algorithm is it does not gives precise continous forecast.

B. Artificial Neural Networks:

A neural network is structured the group of linked similar entities (neurons). The linked entities are used to send signals from one entity(neuron) to the other. Additionally, the linked have it's density to enhance the delivery among neurons. The neurons are not powerful by themselves, however, when connected to others they can perform composite computations. Density on the interconnections are gets updated when the network is trained, hence significant interconnection play more role during the testing phase. Since interconnections do not loop backward or skip more neurons, the network is called feed forward. The power of neural networks comes from the non-linearity of the hidden neurons. In effect, scalable Web API for the testing module's consumption. Brython server-side architecture, which enables to run python in the browser; and RapydScript client-side architecture, which supports compiling python to javascript; have been some of the other options considered during the implementation. However, due to the computation advancements offered by Python over Brython/Rapydscript, the solution has been designed with Python-based training module.

C. Support Vector Machine(SVM):

Support Vector Machine(SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. After that, we perform classification by finding the hyper-plane that differentiate the two classes very well. Support Vectors are simply the co-ordinates of individual observation. Support Vector Machine is a frontier which best segregates the two classes (hyper-plane/ line).

Technical Approach Details

The proposed approach aims at building a browser extension powered by machine learning technique for phishing detection. Furthermore, given the flflexibility of margin and reduced computational complexity offered by SVM, for classifification problem statements, the implementation employs SVM trained persistent model to identify the malicious sites. The extension is packaged to support Chrome browser in specifific, solely by the virtue of its popularity. Additionally, extensions exhibit minimal web-dependence, as it collates multiple fifiles into single fifile for user to download, as one-time activity. A. Browser Extension Schematics The solution deals with training the model with available data-set, using SVM discriminative classififier, followed by passing the persistent model to the extension, which further predicts the authenticity of the user accessed websites and provides alerts to notify the legitimacy of the browsed URL on every page load. The solution integrates Python-based training stage implementation with JavaScript-based testing module. The training component has been designed using Python, so as to make optimal utilisation of the available complex numeric computation libraries. Moreover, given the fact that the testing stage is centric to web-content and feature extraction, and has minimal heavy computation activities associated; the solution does face client-end computation performance lag concerns. During the initial analysis of the project, the team analysed couple of approaches; and weighing the pros, cons and bandwidth of the resources, fifinalised the persistent model passing methodology as the favored methodology. One of the planned approaches aimed at developing Node.js enabled testing component, where the SVM model is structured as scalable Web API for the testing module’s consumption. Brython server-side architecture, which enables to run python in the browser; and Rapydscript client-side architecture, which supports compiling python to javascript; have been some of the other options considered during the implementation. However, due to the computation advancements offered by Python over Brython/Rapydscript, the solution has been designed with Python-based training module. The Chrome extension complies to the

Google norms and, primarily, consists of three main files: manifest.json, content.js, background.js. The manifest file provides all the meta data information about the extension to Chrome browser. Addition-ally, it also specifies all the files and other resources associated to the extension. The content.js file loads on every page in the Chrome browser, post the extension deployment. However, it is an unprivileged module, which has direct access only to the DOM elements and needs supporting files to interact to external APIs and browser user interface manipulation. The supplementary file background.js aids the content script with these interactions, which is termed as message passing.

Experimental Evaluation:

This project compares the performance of all the classifiers described in section 5 on the phishing dataset. We have evaluated these algorithms on 3317 test samples using various performance metrics and this section contains the tabulated results with their graphs. it can be seen that SVM outperforms all the other algorithms based on accuracy in detection of Phishing URL. Also, Figure 3 shows the sensitivity of each classifier. Here sensitivity refers to the classifier's ability to correctly detect phishing URLs. It can be seen that SVM has the highest sensitivity among all the other classifiers. However, in phishing detection, false positives and false negatives are given more attention when studying the total performance (predictive accuracy) of a classifier. That is because false positives are costly than false negatives in the real world. Since we do not want to allow users to access the phishing URLs, false positives are considered to be important while deciding the best classifier. The Figure 4 shows the false positive rates of all the classifiers. It is evident that SVM has the least False positive rate among the three. Hence, SVM works best in classifying the phishing URL from the legitimate URLs.

Conclusion:

Thus to summarize, we have seen how phishing is a huge threat to the security and safety of the web and how phishing detection is an important problem domain. We have reviewed some of the traditional approaches to phishing detection; namely blacklist and heuristic evaluation methods, and their drawbacks. We have tested three machine learning algorithms on the 'Phishing Websites Dataset' from the UCI Machine Learning Repository and reviewed their results. We then selected the best algorithm based on it's performance and built a Chrome extension for detecting phishing web pages. The extension allows easy deployment of our phishing detection model to end users. For future enhancements, we intend to build the phishing detection system as a scalable web

service which will incorporate online learning so that new phishing attack patterns can easily be learned and improve the accuracy of our models with better feature extraction.

Chapter 3

3.1 EXISTING SYSTEM:

- Anti-phishing strategies involve educating netizens and technical defense. In this paper, we mainly review the technical defense methodologies proposed in recent years. Identifying the phishing website is an efficient method in the whole process of deceiving user information
- Along with the development of machine learning techniques, various machine learning based methodologies have emerged for recognizing phishing websites to increase the performance of predictions.
- The primary purpose of this paper is to survey effective methods to prevent phishing attacks in a real-time environment.

PROBLEM STATEMENT:

- A phishing life cycle to clearly capture the phishing problem;
- A survey of major datasets and data sources for phishing detection websites;
- A state-of-the-art survey of machine learning-based solutions for detecting phishing websites

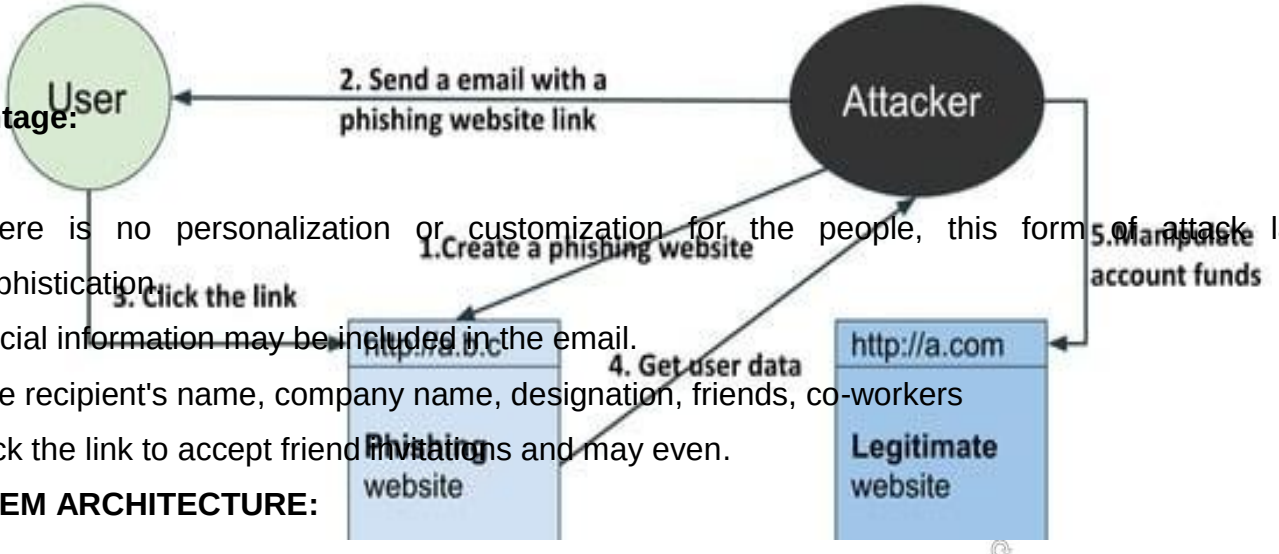
3.2 PROPSOED SYSTEM:

- The most frequent type of phishing assault, in which a cybercriminal impersonates a well-known institution, domain, or organization to acquire sensitive personal information from the victim, such as login credentials, passwords, bank account information, credit card information, and so on.
- Emails containing malicious URLs in this sort of phishing email contain a lot of personalization information about the potential victim.
- To spear phish a "whale," here a top-level executive such as CEO, this sort of phishing targets corporate leaders such as CEOs and top-level management employees
- To infect the target, the fraudster or cyber-criminal employs a URL link. People are sociable creatures who will eagerly.

Advantage:

- There is no personalization or customization for the people, this form of attack lacks sophistication.
- Social information may be included in the email.
- The recipient's name, company name, designation, friends, co-workers
- click the link to accept friend invitations and may even.

3.3 SYSTEM ARCHITECTURE:



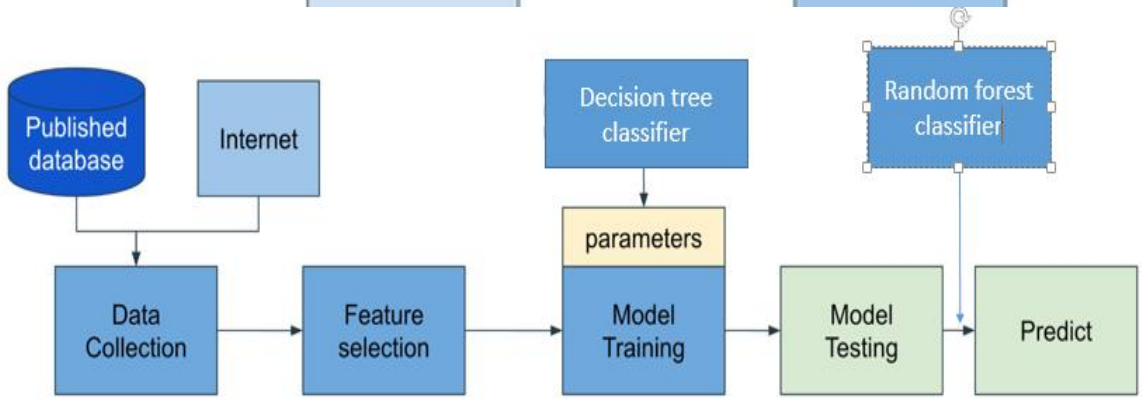


Fig 3.1 BlockDiagram

Fig 3.2 Flow Diagram

3.4 DECISION TREE ALGORITHM:

Introduction:

Tillnowwehavelearnedaboutlinearregression,logisticregression,andtheywereprettyhardtounderstand.
Let'snowstartwithDecisiontree'sandIassureyouthisisprobablytheeasiestalgorithminMachineLearning.There'snotmuchmathematicsinvolvedhere.Sinceitisveryeasytouseandinterpretitisoneofthemostwidelyusedandpracticalmethodsusedin MachineLearning.

Contents

1. WhatisaDecisionTree?
2. ExampleofaDecisionTree
3. Entropy
4. InformationGain
5. WhentostopSplitting?
6. Howtostopoverfitting?

WhatisaDecisionTree?

Itisatoolthathasapplicationsspanningseveraldifferentareas.Decisiontreescanbeusedforclassificationaswellasregressionproblems.The nameitselfsuggests thatitusesaflowchart likeatreestructuretoshowthepredictionsthatresultfromaseriesoffeature-basedsplits.Itstartswitharootnodeandendswithadecisionmadebyleaves.Beforelearningmoreaboutdecisiontreeslet'sgetfamiliarwithsomeofthe terminologies.

RootNodes–
Itisthenodepresentatthebeginningofadecisiontreefromthisnodethepopulationstartsdividingaccordin gtovariousfeatures.

DecisionNodes–thenodesweget aftersplittingtherootnodesarecalled DecisionNode

LeafNodes–thenodeswherefurthersplittingisnotpossiblearecalledleafnodesorterminalnodes

Sub-tree–justlikeasmallportion ofagraphiscalledsub-graph similarlyasub-section ofthisdecisiontreeiscalledsub-tree.

Pruning–isnothingbutcuttingdownsomenodestostopoverfitting.

Exampleofadecisiontree.

Let'sunderstanddecisiontreeswiththehelpofanexample.

Decisiontreesareupside downwhichmeanstherootisatthetopandthenthisrootissplitintovariousseveralnodes.Decisiontrees arenothing butabunch ofif-elsestatementsinlaymanterms.Itchecks iftheconditionistrueandifitisthenitgoestothenextnodeattachedtothatdecision.
Inthebelowdiagram thetreewillfirstaskwhat istheweather? Isitsunny,cloudy,orrainy? Ifyesthenitwillgotothenextfeature whichishumidityandwind.Itwillagaincheckifthereisastrongwindorweak,ifit'saweakwindandit'srain ythenthepersonmaygoandplay.Didyounoticeanything intheaboveflowchart? Weseethatiftheweatherискloudythenwemustgotoplay.Whydidn'titsplitmore?Whydiditstopthere?T oanswer thisquestion, we needtoknow aboutfewmore concepts like entropy, informationgain, andGiniindex.Butinsimpleterms,Icansayherethattheoutput forthetrainingdatasetisalwaysyesforcloudyweather,sincethereisnodisorderlinessherewedon'tn eedtosplitthenodefurther.The goalof machinelearningistodecreaseuncertainty ordisordersfrom thedatasetandforthis,weusedecisiontrees.Nowyoumustbethinkinghowdoiknow whatshould

betherootnode?whatshould bethedecisionnode?whenshould Istopsplitting? Todecidethis,thereisametriccalled“Entropy”whichistheamountofuncertaintyinthedataset.

Entropy

Entropy isnothing butthe uncertainty inour datasetor measure of disorder.Letme trytoexplainthiswiththehelpofanexample.

Suppose youhavea groupof friendswhodecideswhichmovietheycanwatchtogether onSunday.Thereare2choicesformovies,oneis**“Lucy”**andthesecondis**“Titanic”** andnoweveryonehastotelltheirchoice.After everyone givestheir answer we see that“Lucy” gets 4votesand“Titanic” gets 5votes. Whichmoviedowewatchnow?Isn’tithardtochoose 1movienowbecausethevotesforboththemoviesaresomewhatequal.Thisisexactlywhatwecalldiso rderness,
thereisanequalnumberofvotesforboththemovies,andwecan’treallydecidewhichmovieweshould watch.Itwouldhavebeenmucheasieriftthevotesfor“Lucy”were8andfor“Titanic”itwas2.Herewecould easilysaythatthemajorityofvotesarefor“Lucy”henceeveryonewillbewatchingthismovie.

In a decision tree, the output is mostly
“yes” or “no”

TheformulaforEntropyisshownbelow:
Herep+istheprobability
ofpositiveclassp–
istheprobabilityofnegativeclass
Sisthesubsetofthetrainingexample

HowdoDecisionTreesuseEntropy?

Nowweknowwhatentropyisandwhatisitsformula,Next,weneedtoknowthathowexactlydoesitwork inthisalgorithm.
Entropybasicallymeasurestheimpurityofanode.Impurityisthedegreeofrandomness;ittellshowran domourdatais.A**puresub-split**meansthateitheryoushould begetting “yes”,oryoushould begetting“no”.

Suppose feature 1had8yesand4no, after the splitfeature 2get5yesand2nowhereasfeature3gets 3yesand2no.Weseeherethesplitisnotpure,why?Becausewecanstillseesomenegativeclassesinboththefeature.Inordertomakeadecisiontree,weneedtocalculatetheimpurityofeachsplit,andwhenthepurityis100%wemakeitasaleafnode.

Tocheckthe impurityoffeature2andfeature3wewilltakethehelpforEntropyformula.Forfeature2theentropyisasfollows:

Forfeature3,Wecanclearlyseefromthetreeitselfthatfeature2haslowentropy ormorepuritythanfeature3sincefeature2hasmore“yes”anditiseasytomakeadecisionhere.

Alwaysrememberthatthehigher theEntropy,thelowerwillbethepurityandthehigher willbetheimpurity.Asmentionedearlierthe goalof machine learningistodecrease the uncertainty orimpurityinthe dataset,here byusing the entropy we are getting the impurityof a feature or a particular node, we don’tknow if theparententropyortheentropyofaparticularnodehasdecreasedornot.

Forthis,webringanewmetriccalled“Informationgain”whichtellsushowmuchtheparententropy hasdecreasedaftersplittingitwithsomefeature.

InformationGain

Informationgainmeasuresthereductionofuncertainty givensomefeatureanditisalsoa deciding factorforwhichattributeshouldbeselectedasadecisionnodeorrootnode.Itisjustentropy of thefulldataset–entropy of thedatasetgivensomefeature.Tounderstandthisbetterlet’sconsideran

Example:

Suppose ourentirepopulation hasa totalof 30instances. The datasetistopredictwhether thepersonwillgotothegymornot.Let’ssay16peoplegotothegymand14peopledon’t.Nowwehavetwo featurestopredictwhetherhe/shewillgotothegymornot.Feature1is**“Energy”**whichtakestwovalues“high”and“low”.Feature2is**“Motivation”**whichtakes3values“Nomotivation”,“Neutral”and“Highlymotivated”.

Let’sseehowourdecisiontreewillbemade usingthese2features.We’lluseinformationgaintodecidewhichfeatureshouldbetherootnodeandwhichfeatureshouldbeplacedafterthesplit.
Nowweknowwhatentropyisandwhatisitsformula,Next,weneedtoknowthathowexactlydoesitwork inthisalgorithm.

Let’scalculatetheentropy:

Toseetheweightedaverageofentropyofeachnodewewilldoasfollows:
NowwehavethevalueofE(Parent)andE(Parent|Energy),informationgainwillbe:
Ourparententropywasnear0.99andafterlookingatthisvalueofinformationgain,wecansaythattheentropyofthedatasetwilldecreaseby0.37ifwemake“Energy”asourrootnode.
Similarly,wewilldothiswiththeotherfeature“Motivation”andcalculateitsinformationgain.Inthisexample“Energy”willbeourrootnodeandwe’lldothesameforsub-nodes.
Herewecanseethatwhentheenergyis“high”theentropy islowandhence wecansaya personwilldefinitelygotothegym
ifhehashighenergy,butwhatiftheenergyislow?Wewillagainsplitthenodebasedonthenewfeaturewhichis“Motivation”.

Whentostopsplitting?

Youmustbeaskingthisquestion toyourselfthatwhendowestopgrowing ourtree?Usually,real-worlddatasetshavea largenumberof features,whichwillresultina largenumberof splits,whichinturngivesahuge tree.Suchtrees take time to build and canlead to overfitting.That means the tree will give very goodaccuracyonthetrainingdatasetbutwillgivebadaccuracyintestdata.

Therearemanywaystotacklethisproblemthrough hyperparametertuning.
Wecansetthemaximumdepthofourdecisiontreeusing the**max_depth**parameter.The morethevalueof**max_depth**, themorecomplexyourtreewillbe.The trainingerrorwilloff-coursedecreaseifweincreasethe**max_depth**valuebutwhenourtestdata

comesintothepicture,wewillgeta verybadaccuracy.Henceyouneeda valuethatwillnotoverfitaswellasunderfitourdataandforthis,youcanuseGridSearchCV.

Another wayistosettheminimumnumberofsamplesforeachspilt.Itisdenotedby

min_samples_split.

Herewespecifytheminimumnumberofsamplesrequiredtodoaspilt.Forexample,wecanuseaminimumof10samplestoreachadecision.
Thatmeansifanodehaslessthan10samplesthenusingthisparameter,wecanstopthefurthersplittingofthisnodeandmakeitaleafnode.

Therearemorehyperparameterssuchas:

min_samples_leaf–
representstheminimumnumberofsamplesrequiredtobeintheleafnode.Themoreyouincreasethenumber,themoreisthepossibilityofoverfitting.

max_features–ithelpsusdecidewhatnumberoffeaturestoconsider whenlooking forthebest

Pruning:

Itisanothermethodthatcanhelpusavoidoverfitting.Ithelpsinimprovingthe performance of the tree bycutting thenodesorsub-nodes whicharenotsignificant.Itremovesthebrancheswhichhaveverylowimportance.
Therearemainly2waysforpruning:

Pre-pruning –wecanstopgrowing thetreeearlier,whichmeanswecanprune/remove/cut anodeifithaslowimportance**whilegrowing**thetree.
Post-pruning–
onceour**treeisbuilttoitsdepth**,wecanstartpruningthenodesbasedontheirsignificance.

Endnotes

Tosummarize,inthisarticlewelearnedaboutdecisiontrees.Onwhatbasisthetreesplitsthenodesand howtocanstopoverfitting.whylinearregressiondoesn’tworkinthecaseofclassificationproblems. Inthenextarticle,IwillexplainRandomforests,whichisagainanewtechniquetoavoidoverfitting.

RandomForest

Introduction

Randomforestisa**SupervisedMachineLearning Algorithm**that is **usedwidely inClassification andRegression problems**.Itbuildsdecisiontreesondifferentsamplesandtakestheirmajorityvoteforclassification andaverageincaseofregression.

OneofthemostimportantfeaturesoftheRandomForestAlgorithmisthatitcanhandlethedatasetcontain ingcontinuousvariablesasinthecaseofregressionandcategoricalvariablesasinthecaseofclassifica tion.Itperformsbetterresultsforclassificationproblems.

RealLifeAnalogy

Let’s dive into a real-life analogy to understand this concept further.A student named X wants to choose acourseafterhis10+2, andheisconfused aboutthechoiceof coursebasedonhisskillset.Sohedecidestoconsult variouspeople like hiscousins, teachers, parents, degree students, andworking people.He asksthemvariedquestions likewhyheshould choose,jobopportunities withthatcourse,coursefee,etc.Finally,afterconsulting variouspeopleaboutthecoursehedecidestotakethecoursesuggested bymostofthepeople.

Fig 3.3 Real Life Analogy

WorkingofRandomForestAlgorithm

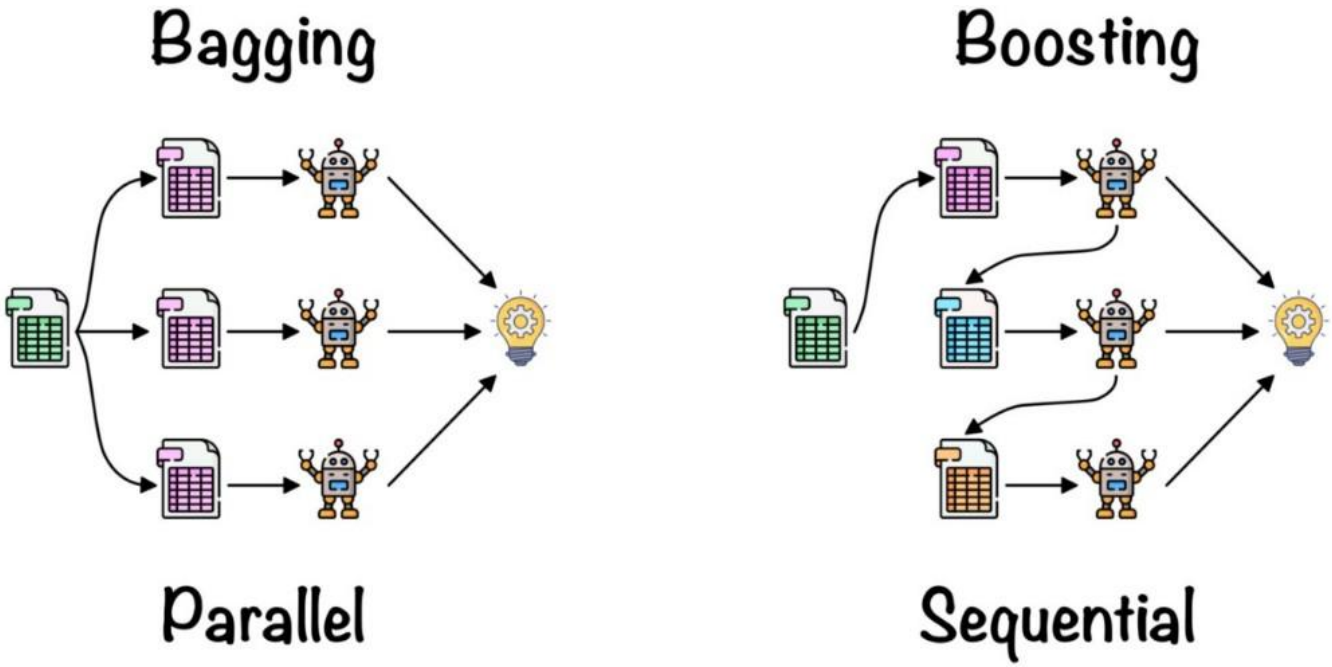
Beforeunderstandingtheworking oftherandomforestwemustlookintotheensembletechnique.Ensemblesimplymeanscombining multiplemodels.Thus acollectionofmodelsisusedtomakepredictionsratherthananindividualmodel.

Ensembleusestwo types of methods:

Bagging–
Itcreatesadifferenttrainingsubsetfromsampletrainingdatawithreplacement&thefinaloutputisbasedon majorityvoting.Forexample,RandomForest.

Boosting–

Itcombinesweaklearnersintostronglearnersbycreatingsequentialmodelssuchthatthefinalmodelhasthe



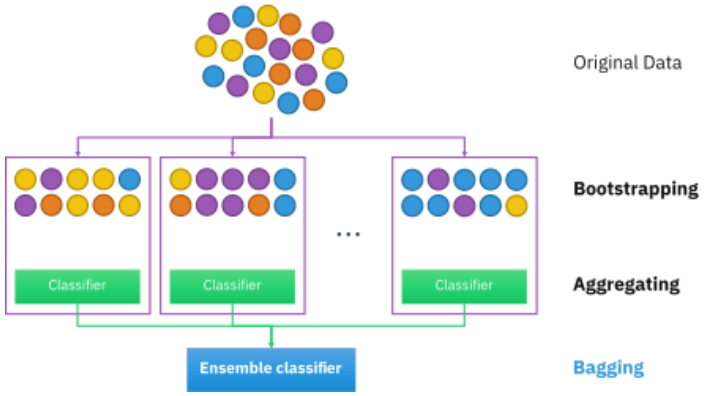
highestaccuracy.Forexample,ADABOOST,XGBOOST.

Fig 3.4 Bagging Parallel & Boosting Sequential

AsmentionedearlierRandomforestworksontheBaggingprinciple.Nowlet'sdiveinandunderstandbagging indetail.

Bagging

Bagging,alsoknownasBootstrapAggregationistheensembletechniqueusedbyrandomforest.Baggingchoosesarandomsamplefromthedataset.Henceeachmodelisgeneratedfromthesamples(BootstrapSamples)providedbytheOriginalDatawithreplacementknownasrowsampling.Thisstepofrowsamplingwithreplacementiscalledbootstrap.Noweachmodelistrainedindependentlywhichgeneratesresults.Thefinaloutputisbasedonmajorityvotingaftercombiningtheresultsofallmodels.Thisstep



whichinvolvescombining allthe

Fig 3.5 Bagging

resultsandgeneratingoutput basedonmajorityvotingisknownasaggregation

Now let's look at an example by breaking it down with the help of the following figure. Here the bootstrap sample is taken from actual data (Bootstrap sample 01, Bootstrap sample 02, and Bootstrap sample 03) with a replacement which means there is a high possibility that each sample won't contain unique data.

Now the model (Model 01, Model 02, and Model 03) obtained from this bootstrap sample is trained independently. Each model generates results as shown. Now Happy emoji is having a majority when compared to sad emoji. Thus based on majority voting final output is obtained as Happy emoji.

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.

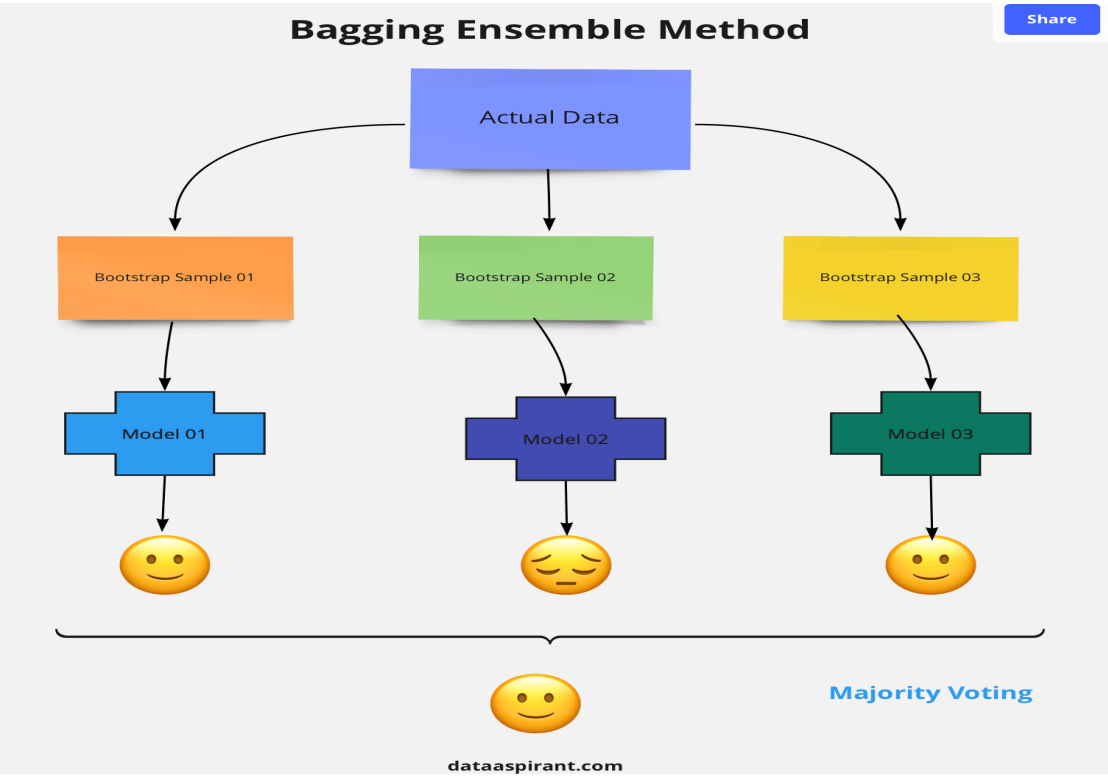


Fig 3.6 Bagging Ensemble Method

For example: consider the fruit basket as the data as shown in the figure below. Now n number of samples are taken from the fruit basket and an individual decision tree is constructed for each sample. Each decision tree will generate an output as shown in the figure. The final output is considered based on majority voting. In the below figure you can see that the majority decision tree gives output as an apple when compared to a banana, so the final output is taken as an apple.

Important Features of Random Forest

Diversity- Not all attributes/variables/features are considered while making an individual tree, each tree is different.

Immune to the curse of dimensionality- Since each tree does not consider all the features, the feature space is reduced.

Parallelization-Each tree is created independently out of different data and attributes. This means that we can make full use of the CPU to build random forests.

Train-Test split- In a random forest we don't have to segregate the data for train and test as there will always be 30% of the data which is not seen by the decision tree.

Stability- Stability arises because the result is based on majority voting/ averaging.

Difference Between Decision Tree & Random Forest

Random forest is a collection of decision trees; still, there are a lot of differences in their behavior.

Decision trees	Random Forest
1. Decision trees normally suffer from the problem of overfitting if it's allowed to grow without any control.	1. Random forests are created from subsets of data and the final output is based on average or majority ranking and hence the problem of overfitting is taken care of.
2. A single decision tree is faster in computation.	2. It is comparatively slower.
3. When a data set with features is taken as input by a decision tree it will formulate some set of rules to do prediction.	3. Random forest randomly selects observations, builds a decision tree and the average result is taken. It doesn't use any set of formulas.

Thus random forests are much more successful than decision trees only if the trees are diverse and acceptable.

Important Hyperparameters

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:

n_estimators– number of trees the algorithm builds before averaging the predictions.

max_features– maximum number of features random forest considers splitting a node.

mini_sample_leaf– determines the minimum number of leaves required to split an internal node.

Following hyperparameters increases the speed:

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.

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.

oob_score – OOB means out of the bag. It is a random forest cross-validation method. In this one-third of the sample is not used to train the data instead used to evaluate its performance. These samples are called out of bag samples.

Hardware requirements:

- System : Pentium i3 Processor.
- Hard Disk : 500 GB.
- Monitor : 15'' LED

- Input Devices : Keyboard, Mouse
- Ram : 2 GB

Software requirements:

- Operating system : Windows 10.
- Coding Language : Python
- Tool:PYCHAM
- Libraries: Open CV,argparse,imutlis,math,dlib,pygam.

Front end

- flask server
- Flask is a lightweight Web Server Gateway Interface WSGI web application framework that was created to make getting started easy and making it easy for new beginners. Flask has its foundation around Werkzeug and Jinja2 and has become one of the most popular Python web application frameworks. Flask is a web framework, it's a Python module that lets you develop web applications easily.

Chapter 4

4.1Machine learning

What are the 7 steps of machine learning?

7 Steps of Machine Learning

- Step 1: Gathering Data. ...
- Step 2: Preparing that Data. ...
- Step 3: Choosing a Model. ...
- Step 4: Training. ...
- Step 5: Evaluation. ...
- Step 6: Hyper parameter Tuning. ...
- Step 7: Prediction.

Introduction:

In this blog, we will discuss the workflow of a Machine learning project this includes all the steps required to build the proper machine learning project from scratch.We will also go over data pre-processing, data cleaning, feature exploration and feature engineering and show the impact that it has on Machine Learning Model Performance. We will also cover a couple of the pre-modelling steps that can help to improve the model performance.

Python Libraries that would be need to achieve the task:

1. Numpy

2. Pandas

3. Sci-kit Learn

4. Matplotlib

Understanding the machine learning workflow

***We can define the machine learning workflow in 3 stages*:**

1. Gathering data

2. Data pre-processing

3. Researching the model that will be best for the type of data

4. Training and testing the model

5. Evaluation

What is the machine learning Model?

The machine learning model is nothing but a piece of code; an engineer or data scientist makes it smart through training with data. So, if you give garbage to the model, you will get garbage in return, i.e. the trained model will provide false or wrong prediction

1. Gathering Data

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.We can also use some free data sets which are present on the internet. Kaggle and UCI Machine learning Repository are the repositories that are used the most for making Machine learning models. Kaggle is one of the most visited websites that is used for practicing machine learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

2. Data pre-processing

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data per-processing and 20% time to actually perform the analysis.

What is data pre-processing?

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn’t feasible for the analysis.Therefore, certain

steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

Why do we need it?

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects. Most of the real-world data is messy, some of these types of data are:

1. **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).

2. **Noisy data:** This type of data is also called outliners, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.

3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

Three Types of Data

1. Numeric e.g. income, age

2. Categorical e.g. gender, nationality

3. Ordinal e.g. low/medium/high

How can data pre-processing be performed?

These are some of the basic pre processing techniques that can be used to convert raw data.

1. **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.

2. **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn't be performed if there are a lot of missing values in the dataset.

3. **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.

4. **Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data.

5. **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0]

Researching the model that will be best for the type of data

Our main goal is to train the best performing model possible, using the pre-processed data.

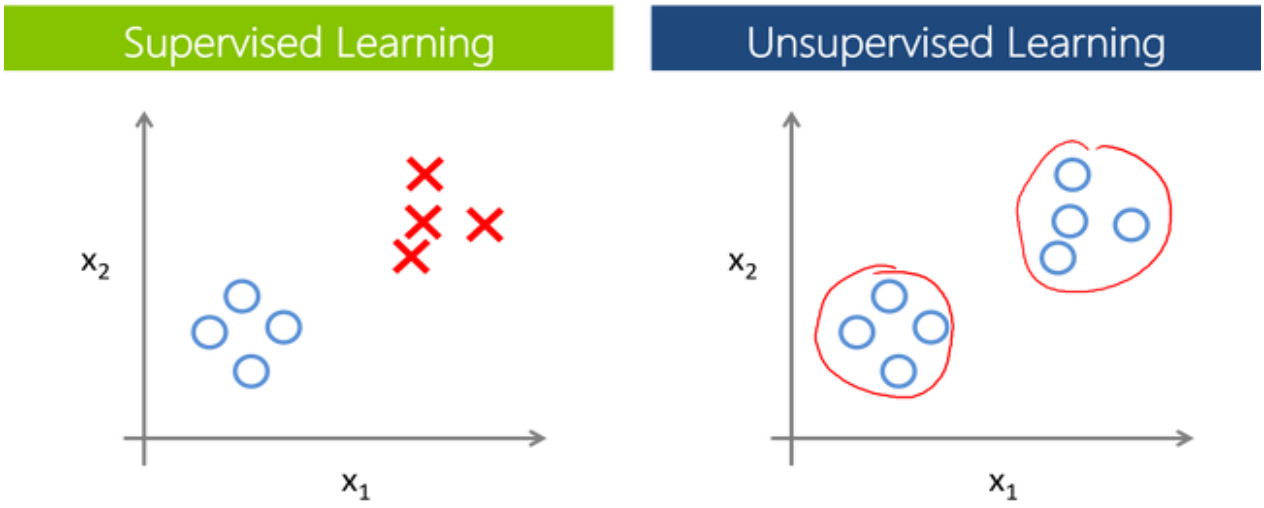


Fig 4.1 Model

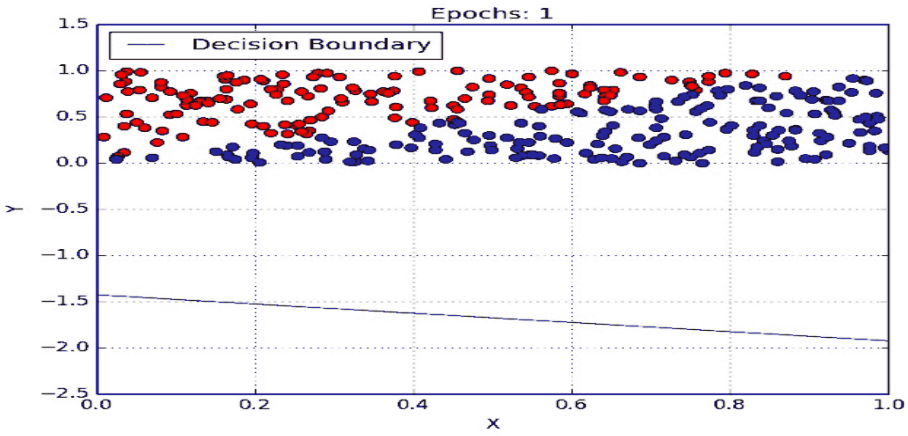
Supervised Learning:

In Supervised learning, an AI system is presented with data which is labelled, which means that each data tagged with the correct label.

The supervised learning is categorized into 2 other categories which are "**Classification**" and "**Regression**".

Classification:

Classification problem is when the target variable is **categorical** (i.e. the output could be classified into classes — it belongs to either Class A or B or something else).A classification problem is when the output variable is a category, such as "red" or "blue" , "disease" or "no disease" or "spam" or "not spam".



As shown in the above representation, we have 2 classes which are plotted on the graph i.e. red and blue which can be represented as 'setosa flower' and 'versicolor flower', we can image the X-

axis as there ‘Sepal Width’ and the Y-axis as the ‘Sepal Length’, so we try to create the best fit line that separates both classes of flowers

These some most used classification algorithms.

K-Nearest Neighbor

Naive Bayes

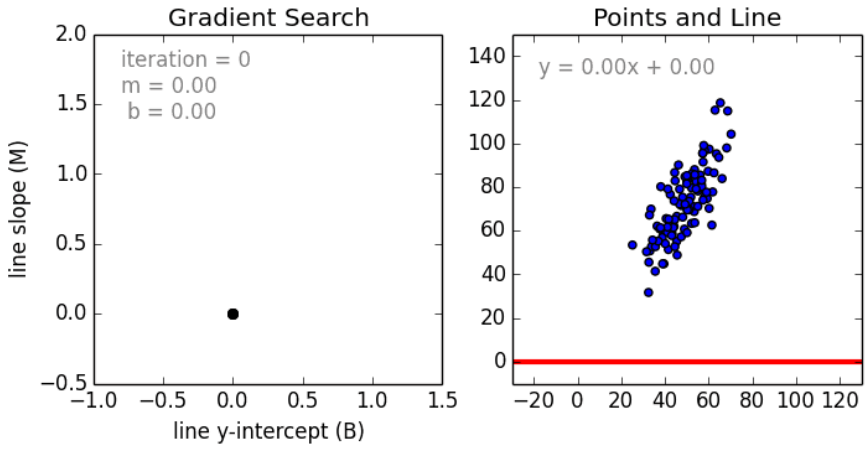
Decision Trees/Random Forest

Support Vector Machine

Logistic Regression

Regression:

While a Regression problem is when the target variable is continuous (i.e. the output is numeric).



As shown in the above representation, we can imagine that the graph’s X-axis is the ‘Test scores’ and the Y-axis represents ‘IQ’. So we try to create the best fit line in the given graph so that we can use that line to predict any approximate IQ that isn’t present in the given data.

These some most used regression algorithms:

Linear Regression

Support Vector Regression

Decision Tress/Random Forest

Gaussian Progresses Regression

Ensemble Methods

Unsupervised Learning:

The unsupervised learning is categorized into 2 other categories which are “Clustering” and “Association”.

Clustering:

A set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

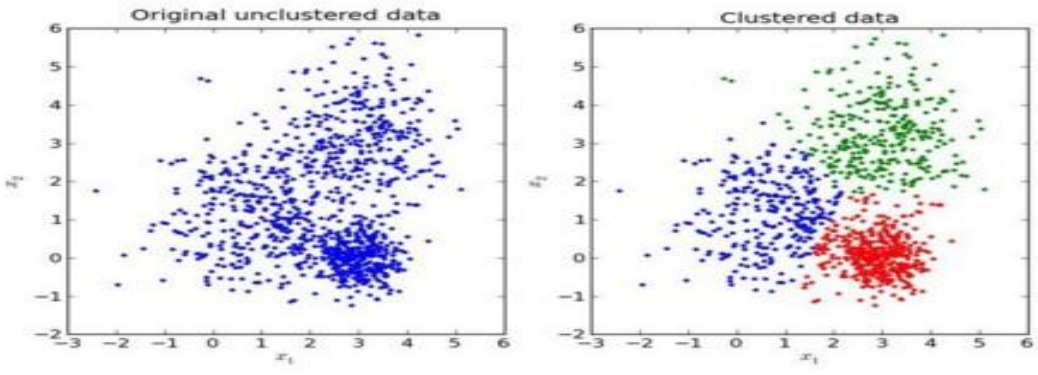


Fig 4.2 Clustering

Methods used for clustering are:

Gaussian mixtures

K-Means Clustering

Boosting

Hierarchical Clustering

K-Means Clustering

Spectral Clustering

Overview of models under categories:

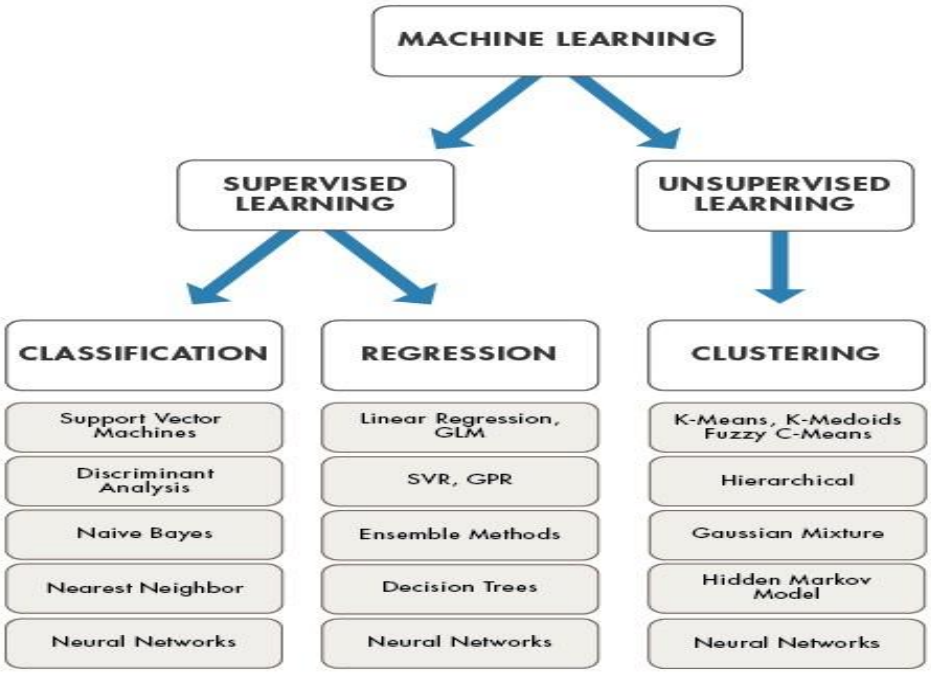


Fig 4.3 ML Overview

4. **Training and testing the model on data**

For training a model we initially split the model into 3 three sections which are ‘Training data’ ,‘Validation data’ and ‘Testing data’.You train the classifier using ‘training data set’, tune the parameters using ‘validation set’ and then test the performance of your classifier on unseen ‘test data set’. An important point to note is that during training the classifier only the training and/or

validation set is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.

Fig 4.4 Training and Testing

Training set:

The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning, that is to fit the parameters of the classifier.

Validation set:

Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. A set of unseen data is used from the training data to tune the parameters of a classifier.

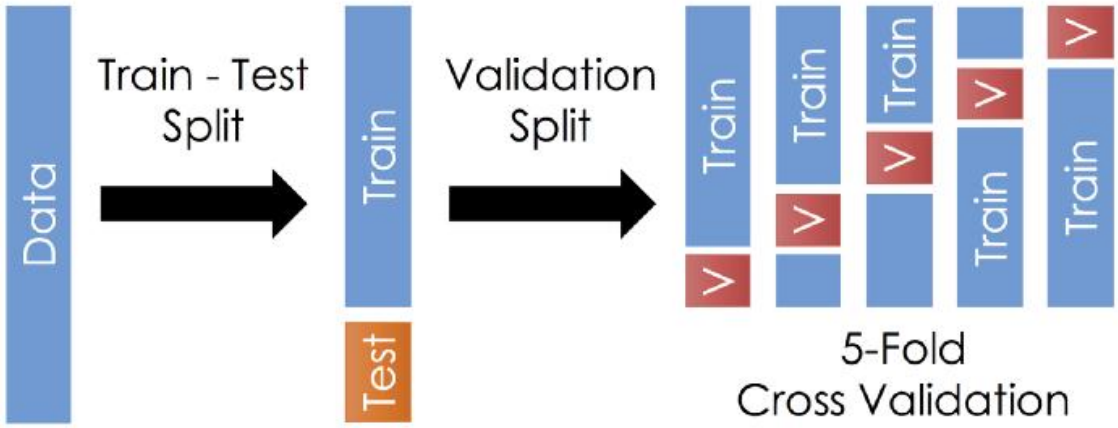


Fig 4.5 Validation Set

Once the data is divided into the 3 given segments we can start the training process.

In a data set, a training set is implemented to build up a model, while a test (or validation) set is to validate the model built. Data points in the training set are excluded from the test (validation) set. Usually, a data set is divided into a training set, a validation set (some people use 'test set' instead) in each iteration, or divided into a training set, a validation set and a test set in each iteration.

The model uses any one of the models that we had chosen in step 3/ point 3. Once the model is trained we can use the same trained model to predict using the testing data i.e. the unseen data. Once this is done we can develop a confusion matrix, this tells us how well our model is trained. A confusion matrix has 4 parameters, which are **'True positives'**, **'True Negatives'**, '**False Positives'** and **'False Negative'**. We prefer that we get more values in the True negatives and true positives to get a more accurate model. The size of the Confusion matrix completely depends upon the number of classes.

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

True positives : These are cases in which we predicted TRUE and our predicted output is correct.

True negatives : We predicted FALSE and our predicted output is correct.

False positives : We predicted TRUE, but the actual predicted output is FALSE.

False negatives : We predicted FALSE, but the actual predicted output is TRUE.

We can also find out the accuracy of the model using the confusion matrix.

Accuracy = (True Positives +True Negatives) / (Total number of classes)

i.e. for the above example:

Accuracy = (100 + 50) / 165 = 0.9090 (90.9% accuracy)

5.**Evaluation**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

To improve the model we might tune the hyper-parameters of the model and try to improve the accuracy and also looking at the confusion matrix to try to increase the number of true positives and true negatives.

6**.Conclusion**

In this blog, we have discussed the workflow a Machine learning project and gives us a basic idea of how a should the problem be tackled.

Chapter 5

SYSTEM DESIGN

***Conclusions*:** In this blog, we have discussed the workflow a Machine learning project and gives us a basic idea of how a should the problem be tackled

Modules:

Data Collection

Data Cleaning

Data Pre-processing

Extraction of Features

Models and Training

Prediction

Working principal

Data Collection :

- The phishing URLs were gathered using the open source tool Phish Tank.
- This site provides a set of phishing URLs in a variety of forms, including csv, json, and others, which are updated hourly.
- This dataset is used to train machine learning models with 5000 random phishing URLs.

***Data Cleaning*:**

- Fill in missing numbers, smooth out creaking data, detect and delete outliers, and repair anomalies to clean up the data.
- Data cleaning is a critically important step in any machine learning project.
- In this module data cleaning is done to prepare the data for analysis by removing or modifying the data that may be incorrect, incomplete, duplicated or improperly formatted.
- In tabular data, there are many different statistical analysis and data visualization techniques you can use to explore your data in order to identify data cleaning operations you may want to perform

Data Pre-processing:

- Data pre-processing is a cleaning operation that converts unstructured raw data into a neat, well-structured dataset that may be used for further research.
- Data pre-processing is a cleaning operation that transforms unstructured raw data into well-structured and neat dataset which can be used for further research.

Extraction of Features*:*

- In the literature and commercial products, there are numerous algorithms and data formats for phishing URL detection.
- A phishing URL and its accompanying website have various characteristics that distinguish them from harmful URLs.
- For example, to mask the true domain name, an attacker can create a long and complicated domain name. Different types of features that are used in machine learning algorithms in the academic study detection process are used

Models and Training*:*

- The data is split into 8000 training samples and 2000 testing samples, before the ML model is trained. It is evident from the dataset that this is a supervised machine learning problem.

- Classification and regression are the two main types of supervised machine learning issues. Because the input URL is classed as legitimate (0) or phishing (0), this data set has a classification problem.
- The following supervised machine learning models were examined for this project's dataset training

Prediction:

- Prediction" refers to the output of an algorithm after it has been trained on a historical dataset and applied to new data when forecasting the likelihood of a particular outcome, such as whether or not a customer will churn in 30 days.
- The algorithm will generate probable values for an unknown variable for each record in the new data, allowing the model builder to identify what that value will most likely be.
- The word "prediction" can be misleading. In some cases, it really does mean that you are predicting a future outcome, such as when you're using machine learning to determine the next best action in a marketing campaign.
- Other times, though, the "prediction" has to do with, for example, whether or not a transaction that already occurred was fraudulent.
- In that case, the transaction already happened, but you're making an educated guess about whether or not it was legitimate, allowing you to take the appropriate action.
- In this module we use trained and optimized machine learning model to predict whether the patient the divers case asking some questions .

Working principal*:*

- A phishing website is a social engineering technique that imitates legitimate webpages and uniform resource locators (URLs).
- The Uniform Resource Locator (URL) is the most common way for phishing assaults to occur.
- Phisher has complete control over the URLs sub-domains.
- The phisher can alter the URL because it contains file components and directories
- This research used the linear-sequential model, often known as the waterfall model.

Chapter 6

Conclusions:

- This survey presented various algorithms and approaches to detect phishing websites by several researchers in Machine Learning.
- On reviewing the papers, we came to a conclusion that most of the work done by using familiar machine learning algorithms like Naïve Bayesian, SVM, Decision Tree and Random Forest.
- Some authors proposed a new system like Phish Score and Phish Checker for detection. The combinations of features with regards to accuracy, precision, recall etc. were used.
- Experimentally successful techniques in detecting phishing website URLs were summarized As phishing websites increases day by day, some features may be included or replaced with new ones to detect them.

REFERENCES:

[1] 'APWG | Unifying The Global Response To Cybercrime' (n.d.) available: https://apwg.org/

[2] 14 Types of Phishing Attacks That IT Administrators Should Watch For [online] (2021)https://www.blog.syscloud.com,available:https://www.blog.syscloud.comtypes-of-phishing/

[3] Lakshmanarao, A., Rao, P.S.P., Krishna, M.M.B. (2021) 'Phishing website detection using novel machine learning fusion approach', in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Presented at the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 1164–1169

[4] H. Chapla, R. Kotak and M. Joiser, "A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier", 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 383-388, 2019, July

[5] Vaishnavi, D., Suwetha, S., Jinila, Y.B., Subhashini, R., Shyry, S.P. (2021) 'A Comparative Analysis of Machine Learning Algorithms on Malicious URL Prediction', in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Presented at the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 1398–1402

[6] Microsoft, Microsoft Consumer safety report. https://news.microsoft.com/en-sg/2014/02/11/microsoft-consumersafety-index-reveals-impact-of-poor-online-safety-behaviours-in-singapore/sm.001xdu50tlxsej410r11kqvksu4nz.

[7] Internal Revenue Service, IRS E-mail Schemes. Available at https://www.irs.gov/uac/newsroom/consumers-warnedof-new-surge-in-irs-email-schemes-during-2016-tax-season-tax-industry-also-targeted.

[8] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S. (2007), A comparison of machine learning techniques for phishing detection.Proceedings of the Anti-phishing Working Groups 2nd Annual ECrime Researchers Summit on - ECrime '07. doi:10.1145/1299015.1299021.

[9] E., B., K., T. (2015)., Phishing URL Detection: A Machine Learning and Web Mining-based Approach. International Journal of Computer Applications,123(13), 46-50. doi:10.5120/ijca2015905665.

[10] Wang Wei-Hong, L V Yin-Jun, CHEN Hui-Bing, FANG Zhao-Lin., A Static Malicious Javascript Detection Using SVM, In Proceedings of the 2nd International Conference on Computer Science and Electrical Engineering (ICCSEE 2013).

[11] Ningxia Zhang, Yongqing Yuan, Phishing Detection Using Neural Net-work, In Proceedings of International Conference on Neural Information Processing, pp. 714–719. Springer, Heidelberg (2004).

[12] Ram Basnet, Srinivas Mukkamala et al, Detection of Phishing Attacks: A Machine Learning Approach, In Proceedings of the International World Wide Web Conference (WWW), 2003.

[13] Sci-kit learn, SVM library. http://scikit-learn.org/stable/modules/svm.html.

[14] Sklearn, ANN library. http://scikit-learn.org/stable/modules/ann.html.

[15] Sclera, Random forest library. http://scikitlearn.org/stable/modules/Randomforesets.html.

APPENDICES

A. SOURCE CODE:

```
# importing the necessary dependencies

from flask import Flask, render_template, request,jsonify

from flask_cors import CORS,cross_origin

import pickle

from features import *



app = Flask(__name__) # initializing a flask app



@app.route('/',methods=['GET'])  # route to display the home page

@cross_origin()

def homePage():

    return render_template("index.html")



@app.route('/predict',methods=['POST','GET']) # route to show the predictions in a web UI

@cross_origin()

def index():
```

```
    if request.method == 'POST':
        try:
            #  reading the inputs given by the user
web_link=str(request.form['link'])
            print(web_link)
            model = int(request.form['model'])
            print(model)

            data=featureExtraction(web_link)
            print(data)

            if model==1:
                filename = 'DecisionTree.pickle'
loaded_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage
                # predictions using the loaded model file
                prediction=loaded_model.predict([data])
print('prediction is', prediction[0])
            else:

                filename = 'RandomForest.pickle'
```

```
loaded_model = pickle.load(open(filename, 'rb')) # loading the model file from the storage

                # predictions using the loaded model file

                prediction=loaded_model.predict([data])

print('prediction is', prediction[0])

            if prediction[0]==1:

                return render_template('result.html', pred =True  ,link=web_link )

            else:

                 return render_template('result.html', pred =False  ,link=web_link )

        except Exception as e:

print('The Exception message is: ',e)

            return 'something is wrong'

    # returnrender_template('results.html')

    else:

        return render_template('index.html')

if __name__ == "__main__":

    #app.run(host='127.0.0.1', port=8001, debug=True)

          app.run(debug=False) # running the app
```

B. SCREENSHOTS:-

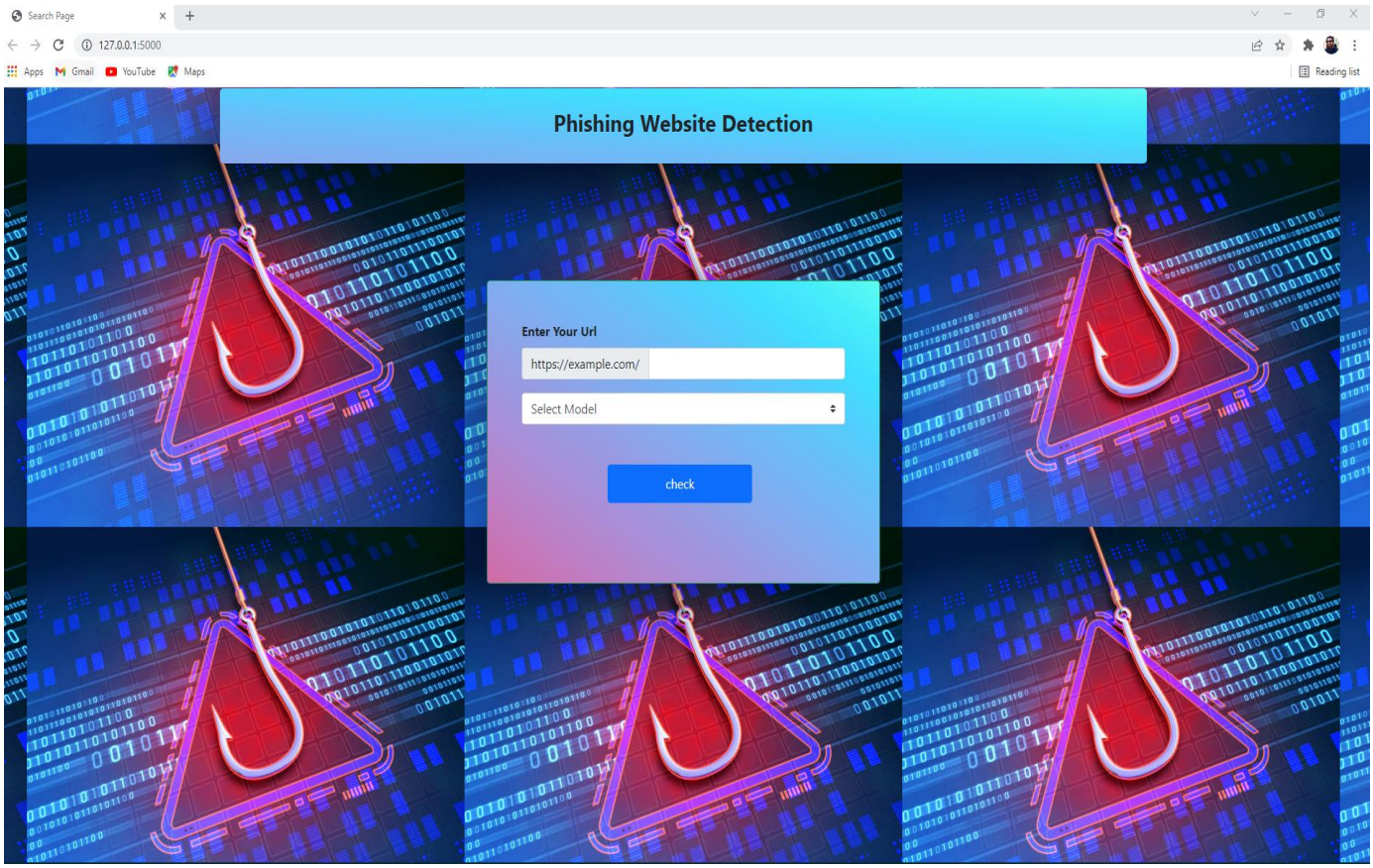


FIG 1 URL Entry

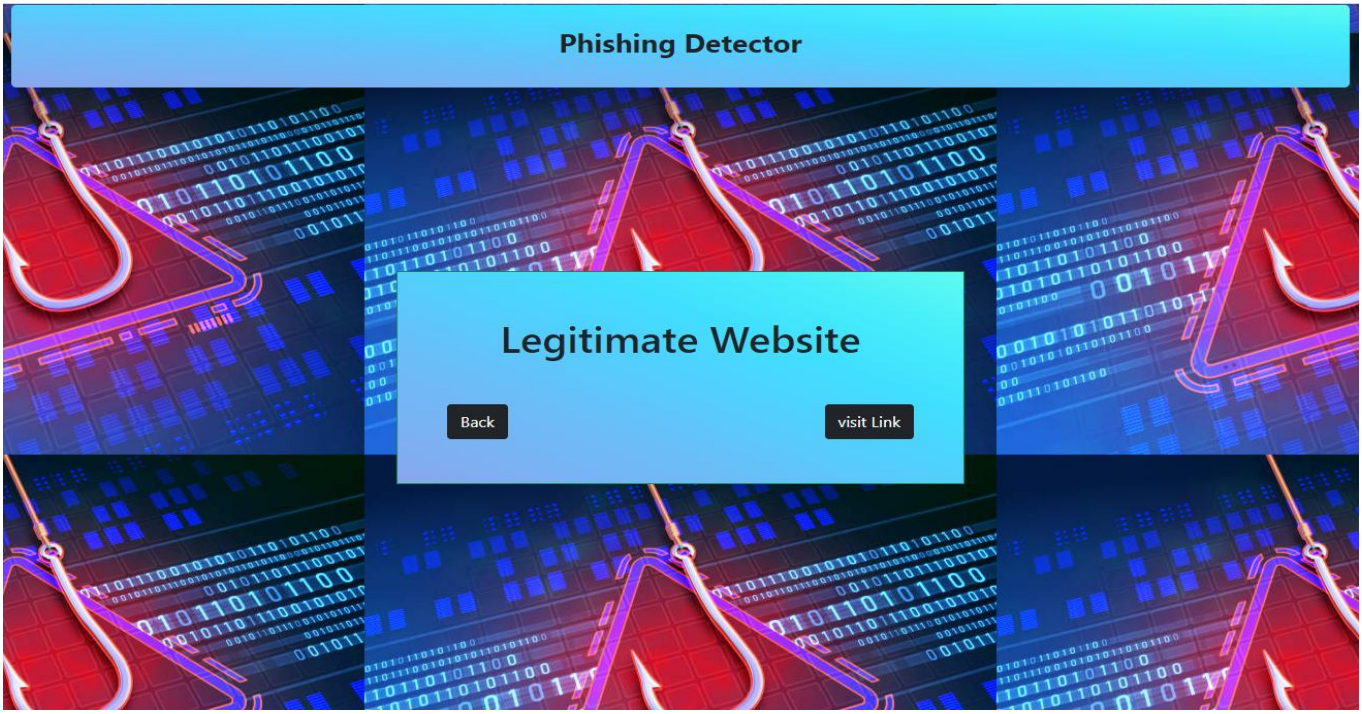


FIG:2 Result

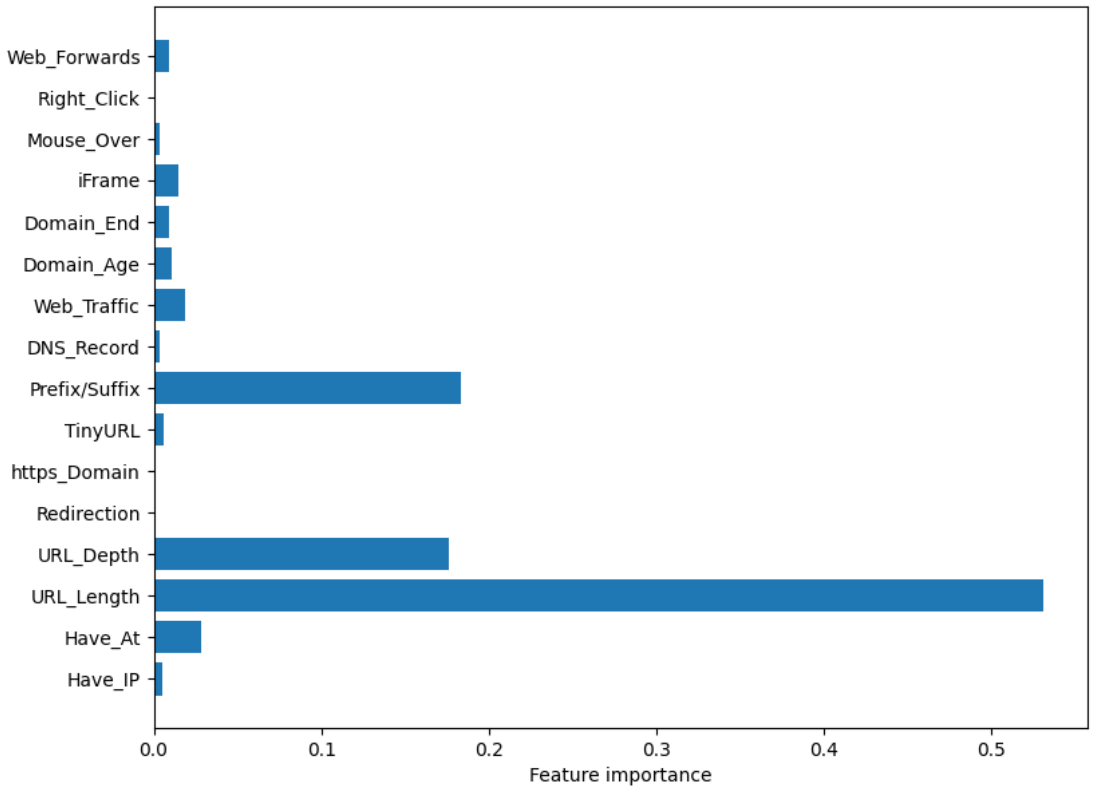


Fig 3 Feature Importance

C. PLAGIARISM REPORT :

PHISHING WEBSITE DETECTION USI...

ORIGINALITY REPORT

6%	4%	5%	2%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	norma.ncirl.ie Internet Source	2%
2	cybersecurity.springeropen.com Internet Source	1%
3	"Table of contents", 2017 IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 15th Intl Conf on Pervasive Intelligence and Computing, 3rd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech), 2017 Publication	1%
4	ijream.org Internet Source	1%
5	www.vahitbicak.com.tr Internet Source	1%
6	Anushka Srivastava, Avishka Agarwal, Gagandeep Kaur. "Novel Machine Learning Technique for Intrusion Detection in Recent	<1%

Network-based Attacks", 2019 4th International Conference on Information Systems and Computer Networks (ISCON), 2019
Publication

7	Nureni Ayofe Azeez, Ahmed Oladapo Lawal, Sanjay Misra, Jonathan Oluranti. "Machine learning approach for identifying suspicious uniform resource locators (URLs) on Reddit social network", African Journal of Science, Technology, Innovation and Development, 2021 Publication	<1 %
8	www.testmagzine.biz Internet Source	<1 %
9	Mahajan Mayuri Vilas, Kakade Prachi Ghansham, Sawant Purva Jaypralash, Pawar Shila. "Detection of Phishing Website Using Machine Learning Approach", 2019 4th International Conference on Electrical, Electronics, Communication, Computer Technologies and Optimization Techniques (ICEECCOT), 2019 Publication	<1 %

Exclude quotes	On	Exclude matches	Off
Exclude bibliography	On		

D. JOURNAL PAPER :

PHISHING WEBSITE DETECTION USING NOVEL MACHINE LEARNING FUSION APPROACH

Aravapalli Sujith Kumar
Student
Department of CSE
Sathyabama Institute of Science and Technology
Chennai, India
sujithkumar112000@gmail.com

Arikatla Gopi Venkata Sudheer
Student
Department of CSE
Sathyabama Institute of Science and Technology
Chennai, India
arikatala@gmail.com

Dr.M.Maheswari
Associate Professor
Department of CSE
Sathyabama Institute of Science and Technology
Chennai, India
maheswari.cse@sathyabama.ac.in

Dr. M.Selvi
Assosiate Professor
Department of CSE
Sathyabama Institute of Science and Technology
Chennai, India
selvi.cse@sathyabama.ac.in

Abstract*: Phishing website is one of the internet securityproblems that target the human vulnerabilities rather thansoftware vulnerabilities. It can be described as the process of attracting online users to obtain their sensitive information such as usernames and passwords. In this paper, we offer an intelligent system for detecting phishing websites. Throughout the long term, numerous analysts have created exceptional techniques for quickly recognizing phishing grounds. Current arrangements can give the best outcomes, yet require more specialized hardware, and it isn't great to be aware of new assaults. Subsequently, it is a straightforward matter in this part to observe a quick limited time technique and zero-speed the executives test day at the phishing site. The site in the host URL contains a ton of data that can be utilized to recognize malware. Machine preparing is an extraordinary method for figuring out how to fish. It additionally wipes out the impediments of the primary technique. We concentrated on the books exhaustively and needed a better approach to recognize phishing grounds utilizing AI methods and AI calculations. The point of this study was to utilize an assortment of information gathered together to prepare the ML model and the profound organization to set up the phishing ground****.***

Keywords—Phishing, Machine learning, Decision tree, Random Forest

1.INTRODUCTION

Phishing and cybercrime are extremely normal in the internet. Phishing assaults have expanded drastically lately, with numerous businesses going to taxpayer driven organizations and monetary foundations on the web. Anglers are bringing in cash and beginning a fruitful business. Different strategies utilized by anglers to target clients incorporate dangers, VOIPs, unlawful organizations, and phony locales. It is extremely simple to make counterfeit destinations that resemble genuine locales as far as construction and content. Indeed, the substance of this webpage will be predictable with their authority site. The reason for these locales is to acquire explicit data, for example, client account number, login ID, secret word to pull out cash, and Visa. At the

point when shoppers answer these inquiries, they are quickly assaulted by phishing. Various

investigations have been led from everywhere the world to forestall phishing assaults. It is feasible to forestall phishing assaults, track down locales and distinguish clients to recognize phishing regions. AI calculations are one of the main abilities in observing a phishing spot. This study checked out various ways of distinguishing fisheries.

2.LITERATURE SURVEY

[1] JianMao, Jingdongian, WenqianTian, S hishiZhu, TaoWei, AiliLi, ZhenkaiLiang 2018
Detecting Phishing Websites via Aggregation Analysis of Page Layouts.

In this article, we expect to further develop fish recognition strategies utilizing machine preparing. Specifically, we offer a learning technique in view of the assortment of comparative page format strategies used to recognize pages. The aftereffects of the review show that our strategy is exact and viable in page search.

[2] Atharva Deshpande, Omkar Pedamkar, Nachiket Chaudhary, Dr. Swapna Borde/ 2021
Detection of Phishing Websites using Machine Learning.

This page records the characters used to learn and get machines. Phishing is renowned for its assailants, since it's simpler to deceive somebody by hitting an awful line than by conquering a safeguard framework. The negative connections in the primary body of the message are expected to demonstrate that these corporate images and other authentic items are being utilized to arrive at harmed associations.
Instructions to learn via vehicle How to realize Ishant Tyagi phishing site; Jatin Shad; Shubham Sharma; Siddharth Gaur; Gagandeep Kaur/2018 This article centers around different AI calculations to decide whether a site is deceitful or real. Machine preparing is famous on the grounds that it can distinguish party

time assaults and is great at conquering new sorts of phishing assaults. In our work, we had the option to precisely decide 98.4% by foreseeing phishing or lawful area.

[3] Ishant Tyagi; Jatin Shad; Shubham Sharma; Siddharth Gaur; Gagandeep Kaur/ 2018.
A Novel Machine Learning Approach to Detect Phishing Websites

One of the best ways of distinguishing these terrible encounters is Machine Learning. This is on the grounds that numerous phishing assaults have normal attributes that can be recognized by AI. In this article, we will analyze the aftereffects of many.

[4] Vahid Shahrivari, Mohammad Mahdi Darabi, Mohammad Izadi/ 2020
Phishing Detection Using Machine Learning Techniques.

Quite possibly the best method for distinguishing these vindictive exercises is to learn Machine Learning. This is on the grounds that many phishing assaults have to do with AI. In this article, we think about the consequences of numerous techniques for AI for phishing site expectation.

[5] Mohith Gowda HR, Adithya MV, Gunesh Prasad S & Vinay S/ 2020.
Development of anti-phishing browser based on random forest and rule of extraction framework

In this article, we present a better approach to handily distinguish a phishing line on the client line. new inquiry In this framework, we use expulsion rules to eliminate site highlights or elements utilizing URL as it were. The rundown is comprised of 30 unique URLs, which will then, at that point, be utilized to decide the reality of the site through the investigation of timberland arranging machines.

[6] Fenny Zalavadia, Akshata Nevrekar, Priyanka Pachpande, Shubhangi Pandey, and Dr. Sharvari Govilkar / 2019.
Detecting Phishing Attacks Using Natural Language Processing and Deep Learning Models.

This strategy will likewise utilize the Advanced Learning System with a short memory time. and how to all the while show the email in the word line and sentence. Phishing assaults sort the email as per explicit signs that give definite data about the beginnings of the fishery. As a rule, most existing frameworks center around the email line contingent upon the head or body part.

[7] Murat Karabatak; Twana Mustafa/ 2018
Performance comparison of classifiers on reduced phishing website dataset

This article analyzes the assortment of UCI phishing news. Its size has been diminished, and contrasting the exhibition of positioning calculations is being considered in the phishing information base. The phishing news program was downloaded from the UCI library for AI. The informational index comprises of 11055 sections and 31 exercises. The presentation of the arranging calculation is currently contrasted with other data on the order calculations. Then, at that point, analyze the arranging exercises of the informational index utilizing the overall calculations gave.

[8] Wesam Fadheel; Mohamed Abusharkh; Ikhlas Abdel-Qader/ 2017

On Feature Selection for the Prediction of Phishing Websites.

In this review, we utilized the Kaiser-Meyer- Olkin test as an examining technique and involved it in an overall fisheries information base, like the UCI site. Moreover, we utilized in reverse and in reverse vector machines as a method for arranging the choice strategy. Our results show that there is a little distinction between the full presentation of the set-up information and the real

execution utilizing the little information gave (around 63% of the first information).

[9] Mehmet, 2021

In this review, suggested a method for phishing detection based on URLs. To compare the results, the researchers utilized eight different algorithms to evaluate the URLs of three separate datasets using various sorts of machine learning methods and hierarchical architectures.

[10] Garera et, 2021

Lassify phishing URLs using logistic regression over hand-selected variables. The inclusion of red flag keywords in the URL.

Features based on Google's Web page and Google's Page Rank qualitys features based on Google's Web page and Google's Page Rank quality.

[11] Vahid ShahrivariVahid Shahrivari, 2020

They used the logistic regression classification method, KNN, Adaboost algorithm, SVM, ANN and random forest.They used the logistic regression classification method, KNN, Adaboost algorithm, SVM, ANN and random forest.

[12] A. Lakshmanarao; P.Surya Prabhakara Rao; M M Bala Krishna, 2019

Phishing website detection using novel machine learning fusion approach.

They used a dataset from UCI and applied a novel fusion classifier and achieved an accuracy of 97%.

[13] R. Kiruthiga, D. Akila, 2018

Phishing Websites Detection Using Machine Learning.

This paper surveys the features used for detection and detection techniques using machine learning.

[14] Oluwatobi Ayodeji Akanbi, ... Elahe Fazeldehkordi ,2019

Website Phishing Detection
The main objective of this chapter is to train and test the individual reference classifiers (C5.0, LR, KNN, and SVM) with the same dataset, design an ensemble.

[15] Aburrouse et al.,2008

This paper proposed model is based on FL operators which is used to characterize me website flushing factors and indicators as fuzzy variables and produces six measures and criteria's of website phishing attack dimensions with a layer structure.

3.PROPOSED SYSTEM

Cybercriminals are a sort of false assault on notable associations, spaces, and associations to acquire individual data, for example, mysteries, passwords, financial balance data, and Mastercard data for casualties. Messages containing malevolent URLs of this kind of phishing email contain explicit data about the person in question. This kind of phishing assaults corporate

leaders, like CEOs and CEOs, to take Balinese chiefs, like CEOs. A backstabber or criminal uses a URL string to assault an objective.

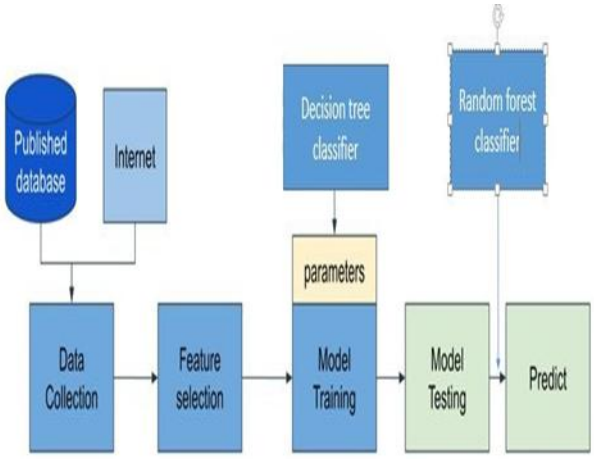


Fig 1: Block Diagram

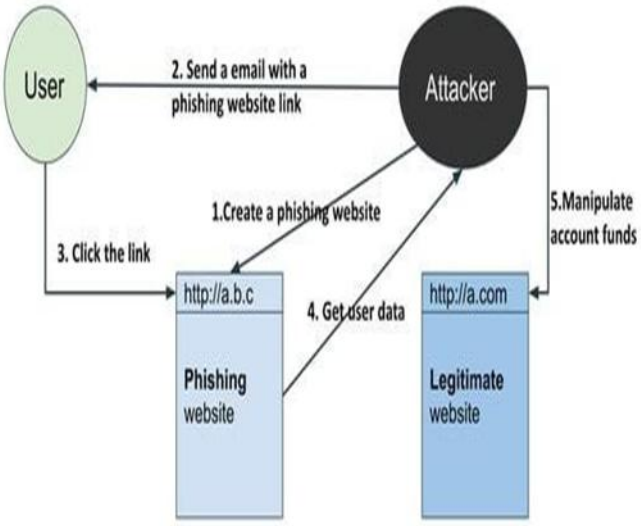


Fig 2: Flow Diagram

3.1 RESEARCH METHODOLOGY

Phishing and person to person communication locales that contain genuine destinations and shared assets (URLs). Uniform Resource Locator (URL) is a typical misguided judgment. Phisher has full oversight over the URL field. The angler can change the URL since it contains portions of the record and envelope.

3.2 DATA COLLECTION:

Phishing URLs are gathered utilizing the Phish Tank discharge instrument. This site gives phishing URLs in an assortment of ways, including csv, json, and other time refreshes. This data bundle is utilized to prepare AI machines and 5,000 novel phishing gear.

3.3 DATA CLEANING

Fill in the missing information, improve on the work process, comprehend the exemptions, erase, and fix uncommon items to clear the information. Information cleaning is a significant stage in all AI exercises. In this module, the cleaning of data is done in anticipation of investigation to eliminate or modify data that might be wrong, fragmented, impersonation, or incorrect. There are numerous procedures for measurable examination and data recovery in bookkeeping pages, and you can utilize them to track down data to discover what information handling exercises you can perform.

3.4 DATA PRE-PROCESSING

The pre-handling and refining process changes unstructured fundamental data into efficient, all-around organized data that can be utilized for inside and out research. Preceding handling, my dad had a cleaning interaction that changed fundamental data that was not organized in a productive and viable manner that could be utilized for top to bottom exploration.

3.5 EXTRACTION OF FEATURES

There are URL search calculations and techniques in the writing and business items. Phishing URLs and going with sites have various attributes that recognize them from vindictive URLs. For instance, concealing a genuine name and an assault can deliver a long and troublesome name. Various kinds of materials utilized in AI calculations as AI illustrations

3.6 MODELS AND TRAINING

Prior to preparing the ML model, the information was partitioned into 8,000 examples and 2,000 estimations. As indicated by the information bundle, this is a machine preparing issue. Arranging and looking are the two most significant parts of AI. Since the URL was entered legitimately or certainly, there was an issue positioning this data. The accompanying instances of AI the board are considered for data bundle project preparing.

- Mod
- A ton of feeling
- Typical backwoods
- Programmed encoder neural organization
- XGBoost
- Support for vector machines

3.7PREDICTION

Prescience is the utilization of algorithmic result and new data subsequent to preparing on a verifiable informational index to foresee when something will occur, for example, deciding if a customer will go inside 30 days. what a worth. "Prescience" might be deceiving. Now and then, this implies that you are making arrangements for the future, like utilizing a machine to decide the following stages in promoting. Notwithstanding, in different cases, "prescience" is related, for instance, assuming an exchange that has as of now occurred is deceitful. For this situation, regardless of whether the exchange is finished, feeling that you have realized whether it is correct or wrong permits you to make the right stride. In this module, we utilize a prepared and created AI strategy to decide whether the patient is posing similar inquiries and posing similar inquiries.

3.8 DECISION TREE CLASSIFIER

The choice tree is utilized adequately to execute and reexamine. Assuming different issues lead to decisions, they generally concentrate fair and square of execution. Make sure to concentrate on the choice tree and follow up to check whether different inquiries get genuine responses the present moment. This strategy is utilized in all investigations to decide the technique for

giving data about the transformative inspiration of logging.

3.9 RANDOM FOREST CLASSIFIER

Ordinary memory is one of the most broadly utilized techniques for AI with respect to rewinding and arranging. A unique memory is an assortment of choice trees, each marginally not the same as the others. The thought behind it is to recall that each tree can work really hard of forecasting, however now and again it will surpass. They are very strong, proficient and don't change the size of the scale, and don't need a lot of information.

3.10 WORKING PRINCIPAL:

A phishing webpage is a specialized method that imitates official sites and shared assets. Connecting assets is a typical method for managing phishing. Phisher has full command over the URL subdomain. Fisher's URL can be changed in light of the fact that it contains document parts and organizers. Albeit the stream strategy is viewed as typical, it works best when it isn't required.

3.11 DATASET

We have gathered data from an open-source Phishing Tank. The information gathered was as csv. There are 18 segments in the data set, and we have refreshed the information base utilizing the main handling strategy. To get data qualities, we utilized a couple of strategies for information age. A couple of drawings and graphs are given to show how the data is introduced and the

way that the qualities are connected. The segment of the space doesn't have anything to do with the AI

model. We presently have 16 person qualities. The usefulness of the information returned by URL phishing is joined into the document evacuation mode without impedance. We need to blend
data to partake in preparing and testing while at the same time arranging conveyance. This wipes out the chance of over-coordination during preparing.

3.12 FUTURE WORK

This study gives an examination of how to gain proficiency with a URL forecast machine. The primary design is to shield security and keep clients from getting to their secret data. It is feasible to decide if a site is legitimate or to utilize a calculation. Studies have shown that XGboost's positioning, contrasted with different models, strongly affected execution. Utilizing the current layout, we can list the URL gave as legitimate or phishing.

4.RESULT AND DISCUSSION

With the last advance in the assessment, we thought about all AI models. The diary level is intended to analyze the presentation of a model. A rundown made to store model workmanship is the mainstays of data. The lower some portion of the code addresses the real consequences of the model. An autonomous example of preparing and exploration information set up reality.

OUTPUT

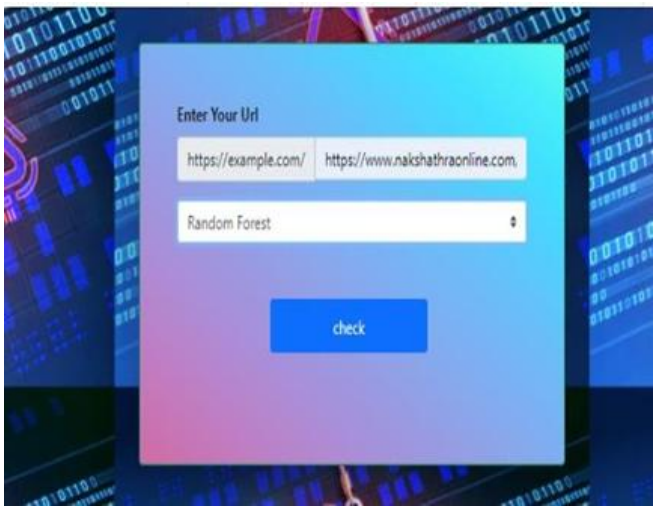


Fig 3: URL Entry

Fig 4: Result

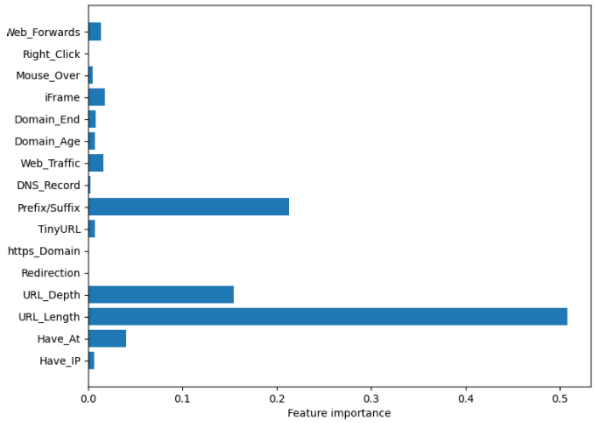


Fig 5: Feature Importance

Feature Importance refers to techniques that calculate a score for all the input features for a given model — the scores simply represent the “importance” of each feature. A higher score means that the specific feature will have a larger effect on the model that is being used to predict a certain variable.

	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	Label
Have_IP	1.000000	-0.011308	-0.076021	-0.030486	-0.008700	-0.001052	-0.025430	-0.023841	-0.011425	0.024279	0.047349	0.018799	-0.004701	0.007251	0.001968	-0.003487	0.074367
Have_At	-0.011308	1.000000	0.067844	0.029944	-0.000297	-0.002151	0.067122	0.016369	0.025073	-0.017002	-0.017072	0.001651	-0.000294	-0.021728	0.004025	-0.030246	0.118419
URL_Length	-0.076021	0.067844	1.000000	0.439378	0.038482	0.007656	-0.005318	-0.146102	-0.019508	0.063717	0.071029	0.028755	-0.039903	-0.068104	0.038033	-0.023651	-0.541267
URL_Depth	-0.030486	0.029944	0.439378	1.000000	-0.040189	-0.000478	0.019880	-0.114919	-0.066073	0.075315	-0.070101	-0.061796	-0.039297	-0.105809	-0.002957	-0.051248	-0.119707
Redirection	-0.008700	-0.000297	0.038482	-0.040189	1.000000	-0.001655	0.026634	-0.025581	-0.027654	0.016784	0.012581	0.025758	-0.012876	-0.017346	0.003896	-0.023193	0.082600
https_Domain	-0.001052	-0.002151	0.007656	-0.000478	-0.001655	1.000000	-0.004456	-0.004534	0.042243	-0.033112	0.018037	0.006852	-0.004472	-0.003778	0.000374	-0.004052	0.014144
TinyURL	-0.023430	0.067122	-0.005318	0.019880	0.026634	-0.004456	1.000000	0.087421	0.059078	0.040888	0.095944	0.006812	-0.062000	-0.054771	0.008339	-0.003508	0.072921
Prefix/Suffix	-0.023841	0.016369	-0.146102	-0.114919	-0.025581	-0.004534	0.087421	1.000000	-0.006793	-0.046843	-0.019954	0.031711	0.050594	0.070263	-0.017527	0.030102	0.382705
DNS_Record	-0.011425	0.025073	-0.019508	-0.066073	-0.027654	0.042243	0.059078	-0.006793	1.000000	0.065776	0.395583	0.162210	0.103266	0.094410	0.008861	0.042350	0.019943
Web_Traffic	0.024279	-0.017002	0.063717	0.075315	0.016784	-0.033112	0.040888	-0.046843	0.065776	1.000000	0.013681	0.015998	0.006990	0.057473	0.051495	0.073455	-0.198793
Domain_Age	0.047349	-0.017072	0.071029	-0.070101	0.012581	0.018037	0.095944	-0.019954	0.395583	0.013681	1.000000	0.329345	-0.034648	-0.018343	0.022232	-0.028868	-0.085077
Domain_End	0.018799	0.001651	0.028755	-0.061796	0.025758	0.006852	0.006812	0.031711	0.162210	0.015998	0.329345	1.000000	-0.042731	-0.007557	0.006449	-0.022273	-0.068556
iFrame	-0.004701	-0.000294	-0.039903	-0.039297	-0.012876	-0.004472	-0.062000	0.050594	0.103266	0.006990	-0.034648	-0.042731	1.000000	0.007077	0.008369	0.617869	0.098446
Mouse_Over	0.007251	-0.021728	-0.068104	-0.105809	-0.017346	-0.003778	-0.054771	0.070263	0.094410	0.057473	-0.018343	-0.007557	0.007077	1.000000	0.007070	0.749077	0.051338
Right_Click	0.001968	0.004025	0.038033	-0.002957	0.003896	0.000374	0.008339	-0.017527	0.008861	0.051495	0.022232	0.006449	0.008369	0.007070	1.000000	0.009008	-0.026467
Web_Forwards	-0.003487	-0.030246	-0.023651	-0.051248	-0.023193	-0.004052	-0.003508	0.030102	0.042350	0.073455	-0.028868	-0.022273	0.617869	0.749077	0.009008	1.000000	-0.041376
Label	0.074367	0.118419	-0.541267	-0.119707	0.082600	0.014144	0.072921	0.382705	0.019943	-0.198793	-0.085077	-0.068556	0.098446	0.051338	-0.026467	-0.041376	1.000000

Fig 6: Correlation Matrix

A correlation matrix is simply a table which displays the correlation. The measure is best used in variables that demonstrate a linear relationship between each other. The fit of the data can be visually represented in a scatterplot.

```
Accuracy of Random forest: 0.815
Accuracy of Decision Tree: 0.856
```

Fig 7: Accuracy Comparision

In this system we have used two algorithms Random Forest and Decision Tree. Each algorithm has it's own accuracy, based on our need we can use anyone of the algorithm.

5.CONCLUSION

This exploration presents calculations and different strategies for recognizing phishing destinations and numerous scientists in Machine Learning. In the wake of checking on the records, we reached the resolution that the greater part of the work was finished utilizing standard AI machines like Naïve Bayesian, SVM, Certificate Tree, and Random Forest. A few creators have requested new frameworks like Phish Score and Phish Checker to discover. A mix of realness, prosperity, and memory was utilized. The Phishing site sums up the fruitful strategies used to track down the URLs of the sites. As the phishing site develops step by step, certain things can be added or supplanted to discover.

REFERENCES

[1] 'APWG | Unifying The Global Response To Cybercrime' (n.d.) available: https://apwg.org/

[2] 14 Types of Phishing Attacks That IT Administrators Should Watch For [online] (2021) https://www.blog.syscloud.com, available: https://www.blog.syscloud.comtypes-of- phishing/

[3] Lakshmanarao, A., Rao, P.S.P., Krishna, M.M.B. (2021) 'Phishing website detection using novel machine learning fusion approach', in 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS),

Presented at the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), 1164–1169

[4] H. Chapla, R. Kotak and M. Joiser, "A Machine Learning Approach for URL Based Web Phishing Using Fuzzy Logic as Classifier", 2019 International Conference on Communication and Electronics Systems (ICCES), pp. 383-388, 2019, July

[5] Vaishnavi, D., Suwetha, S., Jinila, Y.B., Subhashini, R., Shyry, S.P. (2021) 'A Comparative Analysis of Machine Learning Algorithms on Malicious URL Prediction', in 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), Presented at the 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 1398–1402

[6] Microsoft, Microsoft Consumer safety report. https://news.microsoft.com/en- sg/2014/02/11/microsoft-consumersafety- index-reveals-impact-of-poor-online- safety-behaviours-in-singapore/sm.001xdu50tlxsej410r11kqvks u4nz.

[7] Internal Revenue Service, IRS E-mail Schemes. Available at https://www.irs.gov/uac/newsroom/consu mers-warnedof-new-surge-in-irs-email- schemes-during-2016-tax-season-tax- industry-also-targeted.

[8] Abu-Nimeh, S., Nappa, D., Wang, X., Nair, S. (2007), A comparison of machine learning techniques for phishing detection. Proceedings of the Anti-phishing Working Groups 2nd Annual ECrime Researchers Summit on - ECrime '07. doi:10.1145/1299015.1299021.

[9] E., B., K., T. (2015)., Phishing URL Detection: A Machine Learning and Web Mining-based Approach. International Journal of Computer Applications,123(13), 46-50. Doi:10.5120/ijca2015905665.

[10] Erzhou Zhu,Yuyang Chen,Chengcheng Ye,Xuejun Li,Feng Liu, "OFSNN:An Effective Phishing Websites Detection Model Based on Optimal Feature Selection and Neural Network," IEEE Access(Volume:7), pp. 73271-73284, June 2019.

[11] Youness Mourtaji,Mohammed Bouhorma,Alghazzawi, "Perception of a new framework for detecting phishing web pages," Mediterranean Symposium on Smart City Application Article No. 11, Tangier, Morocco, October 2017.

[12] Akihito Nakamura,Fuma Dobashi, “Proactive Phishing Sites Detection,” WI '19 IEEE/WIC/ACM International Conference on Web Intelligence), pp. 443- 448, October 2019.
[13] Ebubekir Büber, ' Phishing URL Detection with M', [Online]. Available: https://towardsdatascience.com/phishing- domaindetection-with-ml-5be9c99293e5 [Accessed: 10- November- 2019].

[14] scikit-learn, Machine Learning in Python, [Online]. Available: https://scikit- learn.org/stable/ [Accessed: 10- November- 2019].
[15] Mohammed Nazim Feroz,Susan Mengel, “Phishing URL Detection Using URL Ranking,” IEEE International Congress on Big Data, July 2015.

Biography

Master Arikatla Gopi Venkata Sudheer graduating from the beloved university Sathyabama Institute of Science and Technology in the stream of Computer Science and Engineering has excelled in this stream. His research interest is Machine Learning, Data Science and Cyber Security.

Master Aravapalli Sujith Kumar graduating from the beloved university Sathyabama Institute of Science and Technology in the stream of Computer Science and Engineering has excelled in this stream. His research interest is Machine Learning, Neural Network and Web Development.

Dr.M.MAheswari, Associate professor in the Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology in Chennai. Her area of specialization is Database Management Systems, Data Mining, Machine Learning, and Data Analytics. She has authored in various reputed conferences and in international journals.

Dr.M.Selvi, Associate professor in the Department of Computer Science and Engineering at Sathyabama Institute of Science and Technology in Chennai. Her area of specialization is Database Management Systems, Data Mining, Machine Learning, and Data Analytics. She has authored in various reputed conferences and in international journals.