VISVESVARAYA TECHNOLOGICAL UNIVERSITY

***"Jnana Sangama"*, Belgavi- 590014**

A Project Report on

"PHISHING WEBSITE DETECTION BASED ON MULTI DIMENTIONAL FEATURES DRIVEN BY DEEP LEARNING"

Submitted By:

AISHA TABASSUM	**4RA14CS003**
ANUSHREE M S	**4RA15CS011**
HAJEERA KHANUM	**4RA16CS029**
POOJA N S	**4RA16CS063**

To the Visvesvaraya Technological University during the academic year 2019-2020
in partial fulfillment for the award of

Bachelor of Engineering
in
Computer Science and Engineering

Under the Guidance of:

Mr. KARTHIK G N
Assistant Professor,
Department of Computer Science & Engineering

Department of Computer Science & Engineering
Rajeev Institute of Technology
Hassan-573201
2019-2020

RAJEEV INSTITUTE OF TECHNOLOGY, HASSAN

(Approved by AICTE, New Delhi and Affiliated to VTU, Belagavi.)

Plot # 1-D, Growth Center, Industrial Area, B-M Bypass Road, Hassan-573201
Ph: (08172)-243180/80/84 Fax: (08172)-243183

Department of Computer Science &Engineering

CERTIFICATE

Certified that the **project** work entitled **"PHISHING WEBSITE DETECTION BASED ON MULTIDIMENTIONAL FEATURES DRIVEN BY DEEP LEARNING"** is carried out by **Ms. AISHA TABASSUM [4RA14CS003]**, **Ms. ANUSHREE M S [4RA16CS011]**, **Ms. HAJEERA KHANUM [4RA16CS029]** and **Ms. POOJA N S [4RA16CS063]** respectively, a bonafide students of **RAJEEV INSTITUTE OF TECHNOLOGY, Hassan** in partial fulfillment for the subject **MAIN PROJECT** in **COMPUTER SCIENCE AND ENGINEERING** of Visvesvaraya Technological University, Belagavi during the year 2019-2020. The project work report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

Mr. KARTHIK G N	**Dr. H.N PRAKASH**	**Dr. A.N RAMAKRISHNA**
Assistant Professor	Head of the Department	Principal
Dept. of Computer Science	Dept. of Computer Science	RIT, HASSAN
& Engineering	& Engineering,	
RIT, HASSAN	RIT, HASSAN	

Name of the examiners — Signature with date

1.

2.

DECLARATION

We, **AISHA TABASSUM, ANUSHREE M S, HAJEERA KHANUM** and **POOJA N S** student of 8th semester B.E, Rajeev Institute Of Technology, Hassan, hereby declare that the project work entitled "**PHISHING WEBSITE DETECTION BASED ON MULTI DIMENTIONAL FEATURES DRIVEN BY DEEP LEARNING**" has been carried out by us under the supervision of Guide **Mr. KARTHIK G N**, Assistant Professor, Department of Computer Science & Engineering, RIT, Hassan which has been submitted in partial fulfillment of the requirements for the award of the Degree of Bachelor of Engineering in Computer Science & Engineering of the Visvesvaraya Technological University, Belagavi is an authentic record of our own independent work carried out by us during the academic year 2019-2020.

We further undertake that the matter embodied in this dissertation has not been submitted to any other Organization/University for any award of degree or certificate.

Place: Hassan

Date:

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned own efforts with success.

We would like to profoundly thank to **Our Management of RIT and** out Director **Dr. Rajeev** for providing such healthy environment for successful completion of main project report.

We would like to express our sincere thanks to our Principal **Dr. A N Ramakrishna,** Rajeev Institute of Technology for his encouragement.

We wish to express our gratitude to **Dr. H N Prakash**, Head of the Department of CSE for providing a good working environment and for his constant support and encouragement.

We would like to express our profound sense of gratitude to our guide **Mr. Karthik G N** Assistant professor, Department of CSE for her guidance, initiative and encouragement that led us to complete this report on main project work.

We would also like to thank to all teaching and non-teaching staff of computer science and engineering department. Who has directly and indirectly helped me in completion of the main project report.

AISHA TABASSUM
ANUSHREE M S
HAJEERA KHANUM
POOJA N S

ABSTRACT

Phishing attacks are growing in the similar manner as e-commerce industries are growing. Prediction and prevention of phishing attacks is a very critical step towards safeguarding online transactions. Data mining tools can be applied in this regard as the technique is very easy and can mine millions of information within seconds and deliver accurate results. With the help of machine learning algorithms like, Random Forest, Decision Tree, Neural network and Linear model we can classify data into phishing, suspicious and legitimate. This can be done based on unique features of phishing websites and user does not need to check individual websites. Rather we can identify and predict phishing, suspicious and legitimate websites by extracting some unique features. The aim of this work was to develop model to safeguard users from phishing attack. The Random Forest, Decision Tree, Linear model and Neural Network algorithms have been used on a phishing dataset. The results of these algorithms have then been compared in terms of accuracy, error rate, precision, and recall.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Phishing attacks are growing constantly as the online transactions and digital media is growing, Anti Phishing Working Group (APWG) reported that, phishing most targeted industries are payment system 45% followed by Financial Institutions 16% , webmail 15%, and Cloud Storage 9% (Phishing Activity Trends Report, 2010). Online community as well as IT industries working with sensitive information are at a big risk due to phishing attack. Phishers make website identical to the actual website to mislead the customers to the forged site in order to steal the important data. Inspite of the fact that today users are conscious and educated of these types of attacks, still large number of users are being cheated under this attack of phishing.

All the people using the web are not experts in recognizing phishing immediately because of too much busy schedule and subconscious state of mind in multitasking era; therefore web users or customers become victim as they share their personal information with the attacker. These days it is very easy to create identical websites by utilizing the source code of the existing HTML code which has led to tremendous growth in phishing. Making small heading changes in the parent code of the website is another method which can be used to fraud the victim by misleading them to phishing websites.

Phishers lure the web users by sending them greetings or lottery offers which attract web customers to share their account information to claim the lottery amount right now without any delay within speculated time or to update some of information's otherwise their account will be blocked. Some of the attackers ask debit, credit and PAN card details for the claim of offer with zero payment once the user provides the details the attackers steal the money immediately. India is highly preferred among hackers because of low awareness and less security measures. As the phishing website attacks mostly target online businesses, banks, Web users, and government, so it has become a national security issue. It is necessary that these attacks are detected at an early stage.But, it is difficult to spot these attacks due to newer methods being used by phishing attackers to commit crime (Barraclough et el., 2013).

In order to make phishing detection successful it should detect with high accuracy and in very less time. Traditional method of phishing detection involved fixed black and white

listing databases. But, these methods are not efficient because a duplicate website can be developed very fast. So most of these methods cannot make an accurate decision dynamically on whether the new website is phishing or legitimate. Hence, number of new phishing websites may be classified as legitimate website. In this situation, it is preferred to develop guidelines to extract specific features from websites and then use them to predict the type of web page.

1.1 General Overview

Phishing is a very popular method used in network attacks and leads to privacy leaks, identity theft and property damage. According to statistics from the Kaspersky Lab, in 2017, 29.4% of user computers were subjected to at least one Malware-class web attack over the year and 199 455 606 unique URLs were recognized as malicious by web antivirus components . In addition, the share of financial phishing increased from47.5% to almost 54% of all phishing detections in 2017. Phishing has become one of the biggest security threats in the Internet.

The spread of phishing is no longer limited to traditional modalities such as e-mail, SMS, and pop-ups. Though the prosperity of the mobile Internet and social networks have brought convenience to users, they have also been employed to spread phishing, such as QR code phishing, spear phishing and spoof mobile applications . In addition, many cunning phishing attacks are hosted on websites that have HTTPS and SSL certificates because many users think that HTPPS websites are likely legitimate .

Phishing presents a diversified development trend, which poses new detection challenges. While phishers are pernicious and hide, security experts and researchers have dedicated many efforts in terms of phishing website detection.Blacklists and whitelists are widely used in phishing website detection. The current common browsers integrate blacklists and whitelists to protect users from phishing attacks. Google provides a blacklist of malicious websites that is continuously updated. Users can check the security of URL links through Google Safe Browsing APIs .

Phishing website detection based on blacklists and whitelists is easy to implement with high running speed and a low false positive rate. However, according to statistics, 47%-83% of phishing websites are added to blacklists after 12 hours, and 63% of phishing websites have a lifespan of only 2 hours; thus, the updating of the blacklist is far behind the

generation of phishing websites. In addition to blacklist and whitelist, machine learning methods are widely used in phishing website detection.

The reason is that malicious URLs or phishing webpages have some characteristics that can be distinguished from legitimate websites, and machine learning can be effective in this regard for processing. Current mainstream machine learning methods of phishing website detection extract statistical features from the URL and the host or extract relevant features of the webpage, such as the layout, CSS, text, and then classify these features. However, these methods only analyze the URL or extract features from a single perspective, which makes it difficult to extract the complete attributes of phishing websites. Moreover, some unreasonable features may reduce the accuracy of detection. The character sequence of the URL is natural, automatically generated feature that avoids the subjectivity of artificially selected features. In addition, it does not require third-party assistance and any prior knowledge about phishing. However, in the process of character sequencing, the difficulty is to effectively extract association and semantic information.

Deep learning

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

Deep learning architectures such as deep neural networks, deep belief networks, recurrent neural networks and convolutional neural networks have been applied to fields including computer vision, speech recognition, natural language processing, audio recognition, social network filtering, machine translation, bioinformatics, drug design, medical image analysis, material inspection and board game programs, where they have produced results comparable to and in some cases superior to human experts.

Phishing attack

Phishing is a type of social engineering attack often used to steal user data, including login credentials and credit card numbers. It occurs when an attacker, masquerading as a trusted entity, dupes a victim into opening an email, instant message, or text message. The recipient is then tricked into clicking a malicious link, which can lead to the installation of malware, the freezing of the system as part of a ransomware attack or the revealing of sensitive information.

An attack can have devastating results. For individuals, this includes unauthorized purchases, the stealing of funds, or identify theft.

Moreover, phishing is often used to gain a foothold in corporate or governmental networks as a part of a larger attack, such as an advanced persistent threat (APT) event. In this latter scenario, employees are compromised in order to bypass security perimeters, distribute malware inside a closed environment, or gain privileged access to secured data.

An organization succumbing to such an attack typically sustains severe financial losses in addition to declining market share, reputation, and consumer trust. Depending on scope, a phishing attempt might escalate into a security incident from which a business will have a difficult time recovering.

Fig 1.1: work flow of phishing attack.

Methods of phishing attack

Different types of phishing attacks have now been identified. Some of the more preferred are listed below.

1. **Deceptive Phishing**. The term ”phishing” originally referred to account theft using instant messaging but the most common broadcast method today is a deceptive email message. Messages about the need to verify account information, system failure requiring users to re-enter their information, imaginary account charges, undesirable account changes, new free services requiring quick action, and many other scams are broadcast to a wide group of recipients with the hope that the inexperience will respond by clicking a link to or signing onto a bogus
2. **Keyloggers and Screenloggers** are particular varieties of malware that track keyboard input and send relevant information to the hacker via the Internet. They can embed themselves into users' browsers as small utility programs known as helper objects that run automatically when the browser is started as well as into system files as device drivers or screen monitors.
3. **Session Hijacking** describes an attack where users' activities are monitored until they sign in to a target account or transaction and establish their authentic credentials. At that point the malicious software takes over and can undertake unauthorized actions, such as transferring funds, without the user's knowledge.
4. **Web Trojans** pop up invisibly when users are trying to log in. They collect the user's credentials locally and transmit them to the attacker.
5. **System Reconfiguration Attacks** modify settings on a user's PC for malicious purposes. For example: URLs in a favourites file might be modified to direct users to look alike websites. For example: a bank website URL may be changed from”bankofabc.com” to”bancofabc.com”.
6. **Data Theft** Unsecured PCs often contain subsets of sensitive information stored elsewhere on secured servers. Certainly PCs are used to access such servers and can be more easily included. Data theft is a widely used approach to business espionage. By stealing confidential communications, design documents, legal opinions, employee related records, etc., thieves profit from selling to those who may want to embarrass or cause economic damage or to competitors.
7. **DNS-Based Phishing** Pharming is the term given to hosts file modification or Domain Name System (DNS)-based phishing. With a pharming scheme, hackers tamper with a company's hosts files or domain name system so that requests for

URLs or name service return a bogus address and subsequent communications are directed to a fake site. The result: users are unaware that the website where they are entering confidential information is controlled by hackers.

8. **Content-Injection Phishing** describes the situation where hackers replace part of the content of a legitimate site with Research Article Volume 7 Issue No.4 International Journal of Engineering Science and Computing, April 2017 10202 http://ijesc.org/ false content designed to mislead or misdirect the user into giving up their confidential information to the hacker. For example, hackers may insert malicious code to log user's credentials or an overlay which can secretly collect information and deliver it to the hacker's phishing server.
9. **Man-in-the-Middle** Phishing is harder to detect than many other forms of phishing. In these attacks hackers position themselves between the user and the legitimate website or system. They record the information being entered but continue to pass it on so that users' transactions are not affected. Later they can sell or use the information or credentials collected when the user is not active on the system.
10. **Search Engine Phishing** occurs when phishers create websites with attractive (often too attractive) sounding offers and have them indexed legitimately with search engines. Users find the sites in the normal course of searching for products or services and are fooled into giving up their information. For example, cheaters have set up false banking sites offering lower credit costs or better interest rates than other banks. Victims who use these sites to save or make more from interest charges are encouraged to transfer existing accounts and misinform into giving up their details.

1.2. Existing System

- Blacklists and whitelists were widely used in phishing website detection. The current common browsers integrate blacklists and whitelists to protect users from phishing attacks. Google provides a blacklist of malicious websites that is continuously updated. Users can check the security of URL links through Google Safe Browsing APIs.
- Phishing website detection based on blacklists and whitelists is easy to implement with high running speed and a low false positive rate. However, according to statistics, 47%-83% of phishing websites are added to blacklists after 12 hours, and 63% of phishing websites have a lifespan of only 2 hours; thus, updating of the

blacklist is far behind the generation of phishing websites. In addition to blacklist and whitelist, machine learning methods are widely used in phishing website detection.

- The reason is that malicious URLs or phishing webpages have some characteristics that can be distinguished from legitimate websites, and machine learning can be effective in this regard for processing.
- Current mainstream machine learning methods of phishing website detection extract statistical features from the URL and the host or extract relevant features of the webpage, such as the layout, CSS, text , and then classify these features.
- However, these methods only analyze or extract features from a single perspective, which makes it difficult to extract the complete attributes of phishing websites. Moreover, some unreasonable features may reduce the accuracy of detection.
- The character sequence of the URL is natural, automatically generated feature that avoids the subjectivity of artificially selected features. In addition, it does not require third-party assistance and any prior knowledge about phishing.
- However, in the process of character sequencing, the difficulty is to effectively extract association and semantic information.
- To address these problems, we propose a multidimensional feature phishing detection approach based on a fast detection method by using deep learning (MFPD).

1.3. Proposed System

The aim of this project was to develop model to safeguard users from phishing attack. We develop a website it will check the entered URL is legitimated or not, in these way safeguard the users from phishing attack. We propose a multidimensional feature phishing detection approach based on fast detection method by using deep learning. A combination or hybrid machine learning algorithm can also be implemented to improve success rate and minimize false rate.

1.4. Problem Statement

- As a crime of employing technical means to steal sensitive information of users, phishing is currently a critical threat facing the Internet, and losses due to phishing are growing steadily.
- Feature engineering is important in phishing website detection solutions,, but the accuracy of detection critically depends on prior knowledge of features.
- Moreover, although features extracted from different dimensions are more comprehensive, a drawback is that extracting these features requires a large amount of time.
- To address these limitations, we propose a multidimensional feature phishing detection approach based on a fast detection method by using deep learning (MFPD).

1.5. Objectives

We propose a multidimensional feature phishing detection approach based on a fast detection method by using deep learning (MFPD).

- In the first step, character sequence features of the given URL are extracted and used for quick classification by deep learning, and this step does not require third-party assistance or any prior knowledge about phishing.
- In the second step, we combine URL statistical features, webpage code features, webpage text features and the quick classification result of deep learning into multidimensional features.
- The approach can reduce the detection time for setting a threshold. Testing on a dataset containing millions of phishing URLs and legitimate URLs, the accuracy reaches 98.99%, and the false positive rate is only 1.01%.
- By reasonably adjusting the threshold, the experimental results show that the detection efficiency can be improved.

1.6 . Applications

1. **Activating multi-factor authentication:**
    - There are chances that a phishing attack could succeed, resulting in the theft of an employee's username and password.
    - To mitigate that risk, deploy multi-factor authentication to ensure that, even if the credentials are stolen, the malicious actors won't be able to access your applications, services, and sensitive data.
2. **Deployment of multiple layers of security defenses:**
    - Working on the principle that malicious actors will do their utmost to bypass your security and will continuously modify their tactics, having a defense in depth security strategy is best practice.
    - Start with having an email filter that scans all of the incoming emails to your business; this will block a decent portion of phishing attempts. Then, have an endpoint Anti-Virus product that also includes phishing protection.
    - Finally, deploy a solution that looks at the outgoing web request in the event that a user clicks on a malicious link. This can either be a DNS or a proxy based solution.

CHAPTER 2

LITERATURE SURVEY

1. Shrivas, A. K., &Suryawanshi, R, Decision Tree Classifier for Classification of Phishing Website with Info Gain Feature:

They collected phishing data set from UCI repository and implemented the phishing data set on rapid miner tool and compared decision tree, random tree, random forest algorithms for classification of phishing and non-phishing. Accuracy obtained from decision tree was 91.8% which was the best as compared to other algorithms random tree 66.7%, random forest 78.8% and decision stump 84%. Compared various algorithms like random forest, support vector machine, decision tree on WEKA open source tool. This work reported that Random Forest is quicker, robust and more accurate as compared to KNN, SVM Rotation forest and Decision Tree. Random forest yielded best results here based on accuracy 97.36%.

2. Hodzic, A., and Kevric J, "Comparison of Machine Learning Techniques in Phishing Website Classification":

Hodzic and kevriccompared the algorithms multilayer perceptron (MLP), decision tree, random forest, C4.5, rotation tree (REP tree) etc. All the experiments were conducted in WEKA tool and this work reported that Rotation tree achieved overall best accuracy of 89.1% compared to other algorithms.

3. Mohammad, R. M., Thabtah, F., &McCluskey, L, "Intelligent rule-based phishing websites classification":

Mohammad in his research proposed an Artificial Neural Network (particularly self structuring neural networks) based intelligent model for predicting phishing attacks. The authors were able to atomise phishing website detection with frequent change in phishing websites using 17 different features. Aburrous et al. purposed an Intelligent system to detect phishing in e-banking where they combined fuzzy logic model with machine learning algorithms to detect phishing websites. They differentiated between

different types of phishing websites using 10 fold cross validation and achieved 86.38% accuracy, which is very low.

4. Kalaiselvan, O. &Edwinraja, S,"Predicting Phishing Websites using Rule Based Techniques":

In their paper (Kalaiselvan et. al) collected phishing dataset from the Phistank website and compared the algorithms C4.5, SVM, Naïve and ZeroR, to classify phishing dataset into phishing and legitimate. Here the accuracy of developed methods was assessed after applying the 10 fold cross-validation and Naïve Bayes algorithm was found to perform better than other algorithm. In their research paper Support Vector Machine, Gaussian and NMC classifiers have been employed along with fuzzy logic. Fuzzy based detection system provides effective aid in detecting phishing websites. It successfully resulted in low false positive and high true positive for classifying phishing websites. A methodology to detect phishing website based on machine learning classifiers is presented in (Ali et. al, 2017) which uses a wrapper features selection method. Some common supervised machine learning techniques have been used by authors here to accurately detect phishing websites and they found that wrapper based algorithm performs better as compared to normal method.

5. Chen, H., Vasardani, M., &Winter, S, Geo-referencing Place from Everyday Natural Language Descriptions:

Chen H purposed method for concocted spoof detection using different algorithms as Bayesian Network, C4.5, Logit Regression, Naïve Bayes, Neural Network and SVM (linear composite, linear, polynomial, RBF kernels). The authors achieved an accuracy of 92.56% among 900 legit concocted, and spoof e-commerce websites.

6. M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index":

He proposed a lightweight phishing website detection method that used only six URL features, namely, the URL size, the number of hyphens, the number of dots, the number of numeric characters plus a discrete variable that corresponds to the presence of an IP address in the URL, and finally, the similarity index. The features extracted are completely based on URLs, and because of their low features, the detection speed is fast. However, the amount of experimental data was relatively small. we have presented a phishing websites detection

system 100% based on the URL. Our system has been tested on a database of 2000 records formed from legitimate websites and their phishing counterparts; our system has given very satisfactory and encouraging results precisely a 95.80% recognition rate as shown by the results of the tests. The used approach in this system rests on a powerful tool of AI precisely support vector machine, provided with the Hamming distance between the phishing website and its target and five other features extracted from the URL as input. The advantage of this system is its lightness and it can be incorporated into smartphones and tablets.

We see as perspective to this work to test this system constantly on gigantic phishing websites database to improve it if this is mandatory. We will also use the methods of probabilistic prediction on the phishing websites to predict potential target website based solely on the URL of the phishing website.

7. E. Buber, Ö. Demir and O. K. Sahingoz, "Feature selections for the machine learning based detection of phishing websites":

This paper focuses on detecting phishing website URLs with domain name features. Web spoofing attack categories content-based, heuristic-based and blacklist-based approaches are explained and the proposed model PhishChecker is developed with the help of Microsoft Visual Studio Express 2013 and C# language. Dataset used from Phishtank and Yahoo directory set and obtained an accuracy of 96%. This paper checks only the validity of URLs.

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 PhishScore and PhishChecker for detection. The combinations of features with regards to accuracy, precision, recall etc. were used. Experimentally successful techniques in detecting phishing website URLs. As phishing websites increases day by day, some features may be included or replaced with new ones to detect them.

8. S. Marchal, K. Saari, N. Singh and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets":

Marchal et al. proposed a scalable and languageindependent phishing website detection method. In terms of URL and HTML, 212 features were selected; Gradient Boosting was used to detect phishing websites and yielded a high accuracy. Phishing detection based on the

combined features more fully represents the website, and therefore, the detection effect is better. However, it is necessary to download a webpage or obtain data from a third-party website, and there some issues remain, namely, that the feature extraction is complicated, and real-time detection cannot be satisfied.

9.Bhagyashree E. Sananse, Tanuja K .Sarode: Phishing URL Detection: A Machine Learning and Web Mining-based Approach:

In this paper, a system has been proposed that uses lexical features, WHOIS features, PageRank and Alexa rank and PhishTank-based features for Random Forest algorithm to classify phishing URLs. It has been demonstrated that by applying web mining heuristics on Random Forest algorithm, a precision of more than 90% has been achieved and FNR and FPR rates less than 1%. But in case of Content-based algorithm the precision achieved was less than 65%

A feature-based approach has been proposed for classification of URLs into phishing or non phishing based on the details available on the URLs. This problem is considered as a binary classification problem where phishing URLs are labelled as positive class and benign URLs are labelled as negative class. Firstly phishing and legitimate sites are collected to build the dataset. Then a batch of code is run to collect a number of features on the URLs. Then two algorithms are applied as follows:

1. Random Forest algorithm, which is one of the most efficient machine learning algorithm to build prototypes from training data, which consists of pairs of features values and class labels. The prototypes are then fed with separate set of testing data and the data instance of the predicted class is compared with the actual class of data
2. . Content-based algorithm, (works on the publicly available data on the URLs) which focuses on the important features that distinguish phishing sites from legitimate ones.

As the phishing URL detection problem is binary classification problem, every URL falls into one of four possible categories: true positive (TP, correctly classified phishing URL), true negative (TN, correctly classified nonphishing URL), false positive (FP, non-phishing URL wrongly classified as phishing), and false negative (FN, phishing URL wrongly classified as non- phishing). Standard measures such as false positive rate (FPR), false negative rate (FNR), precision, recall, and F-measure were determined using the following equations

Random Forest classifier has been trained using a set of 500 URLs and tested the classifier using a set of 100 URLs. the output when legitimate URL is fed as input to the system. In www.youtube.com is fed as input to the system and verified for its result. The output as “non-phishing” by both the algorithms, i.e. Random Forest algorithm and Content-based algorithm.

As future work, there is a need to work on selection of more efficient features for Content-based algorithm to increase the precision and decrease the FNR and FPR. Also webpage content based features can be integrated to make the system more robust.

10. MonaliDeshmukh1 , Shraddha K. Popat2 UG Student1 , Assistant Professor2 Department of Computer Engineering D .Y. Patil College of Engineering, Pune, India

The paper has able to conclude that the majority of the ant phishing techniques specialize in contents of net age, universal resource locator and email. Character based ant phishing approach could end in false positive however content based approach ne’er leads to false positive. Attribute based mostly approach think about most major areas susceptible to phishing thus it will be best anti-phishing approach which will notice referred to as well as unknown phishing attack. Identity based mostly anti-phishing approach could fails if phisher gets physical access to client’s laptop. Anti phishing techniques have been discussed. Some techniques having advantages and disadvantage.Mobishis lightweight scheme use to protect phishing attack on mobile computing platform.

CHAPTER 3

BACKGROUND

3.1. History

Phishers According to APWG, the term phishing was coined in 1996. due to social engineering attacks against America On-line(AOL) accounts by online scammers. The term phishing comes from fishing in a sense that fishers(i.e. attackers) use a bait (i.e. socially-engineered messages) to fish (e.g. steal personal information of victims). However, it should be noted that the theft of personal information is mentioned here as an example, and that attackers are not restricted. The origins of the ph replacement of the character f in fishing is due to the fact that one of the earliest forms of hacking was against telephone networks, which was named Phone Phreaking. As a result, ph became a common hacking character replacement of f. According to APWG, stolen accounts via phishing attackswere also used as a currency between hackers by 1997 to trade hacking software in exchange of the stolen accounts. Phishing attacks were historically started by stealing AOL accounts, and over the years moved into attacking more profitable targets, such as on-line banking and e-commerce services. Currently, phishing attacks do not only target system end-users, but also technical employees at service providers, and may deploy sophisticated techniques such as MITB attacks.

•Phishing Motives

The primary motives behind phishing attacks, from an attacker's perspective, are:

• Financial gain: phishers can use stolen banking credentials to their financial benefits.

• Identity hiding: instead of using stolen identities directly, phishers might sell the identities to others whom might be criminals seeking ways to hide their identities and activities (e.g. purchase of goods).

• Fame and notoriety: phishers might attack victims for the sake of peer recognition.

•Importance

According to APWG, phishing attacks were in a raise till August, 2009 when the all-time high of 40,621 unique 3 phishing reports were submitted to APWG. The total number of submitted unique phishing websites that were associated with the 40,621 submitted reports in August, 2009 was 56,362. As justified by APWG, the drop in phishing campaign reports in the years 2010 and 2011 compared to that of the year 2009 was due to the disappearance of the Avalanche gang 4 which, according to APWG's 2 nd half of 2010 report, was responsible for 66.6% of world-wide phishing attacks in the 2 nd half of 2009. In the 1 st half of the year 2011, the total number of submitted phishing reports to APWG was 26,402, which is 35% lower than that of the peak in the year 2009.

However, according to APWG, the drop in phishing attacks was due to the switch in the activities of the Avalanche gang from traditional phishing campaigns into malware-based phishing campaigns. In other words, the Avalanche gang did not stop phishing campaigns but rather switched their tactics toward malware-based phishing attacks(which still requires electronic communication channels and social engineering techniques to deliver malware).

Among the various types of malware that are used in phishing attacks, Trojan horses software seem to be in a raise, and are the most popular type of malware deployedby phishing attacks. According to APWG, Trojans software contributed 72% of the total malware detected in the 1 st half of 2011, from the previous value of 55% in the 2 nd half of 2010.

It is also important to note that although the number of phishing attack reports dropped since the peak in 2009, the number of phishing attack reports are still high ,compared to that of the 2 nd half of 2008 which faced an average of 28,916 unique reports, and ranged between 22,000 and 26,000 of unique reports each month in the 1st half of 2011.

On the other hand, the 2 nd half of 2011 saw a raise in phishing reports and websites, which seems to be correlated with holidays season as depicted2. Which is further amplified when knowing that each phishing campaign can be sent to thousands or even millions of users via electronic communication channels. The year 2011 saw a number of notable spear phishing attacks against well known security firms such as RSA and HB Gary, which resulted in further hacks against their clients such as RSA's client Lockheed Martin.

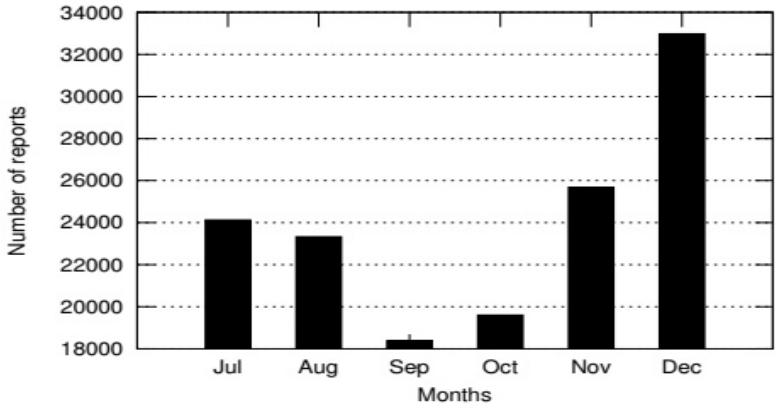

Fig. 3.1: Total number of submitted unique phishing reports

This shows that the dangers of phishing attacks, or security vulnerabilities due to the human factor, are not limited to the naivety of end-users since technical engineers can also be victims. Minimizing the impact of phishing attacks is extremelyimportant and adds great value to the overall security of an organization.

•Challenges

Because the phishing problem takes advantage of human ignorance or naivety with regards to their interaction with electronic communication channels (e.g. E-Mail, HTTP, etc. . .),it is not an easy problem to permanently solve. All of the proposed solutions attempt to minimize the impact of phishing attacks.

From a high-level perspective, there are generally two commonly suggested solutions to mitigate phishing attacks:

• User education; the human is educated in an attempt to enhance his/her classification accuracy to correctly identify phishing messages, and then apply proper actions on the correctly classified phishing messages, such as reporting attacks to system administrators.

• Software enhancement; the software is improved to better classify phishing messages on behalf of the human, or provide information in a more obvious way so that the human would have less chance to ignore it.

The challenges with both of the approaches are:

• Non-technical people resist learning, and if they learn they do not retain their knowledge permanently, and thus training should be made continuous. Although some researchers agree that user education is helpful, a number of other researchers disagree . Stefan Gorling says that: “this is not only a question of knowledge, but of utilizing this knowledge to regulate behavior. And that the regulation of behavior is dependent on many more aspects other than simply the amount of education we have given to the user”

• Some software solutions, such as authentication and security warnings, are still dependent on user behavior. If users ignore security warnings, the solution can be rendered useless.

• Phishing is a semantic attack that uses electronic communication channels to deliver content with natural languages (e.g. Arabic, English, French, etc. . .) to persuade victims to perform certain actions. The challenge here is that computers have extreme difficulty in accurately understanding the semantics of natural languages. A notable attempt is E-mail-Based Intrusion Detection System (EBIDS) , which uses Natural Language Processing (NLP) techniques to detect phishing attacks, however its performance evaluation showed a phishing detection rate

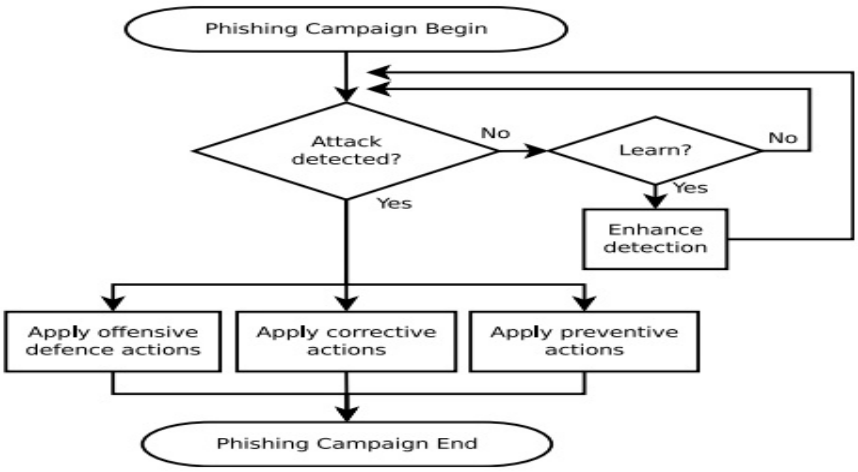

Fig 3.2: The life-cycle of phishing campaigns from the perspective of anti- phishing techniques.

of only 75%. In our opinion, this justifies why most well-performing phishing classifiers do not rely on NLP techniques.

3.2 MITIGATION OF PHISHING ATTACKS : AN OVERVIEW

Due to the broad nature of the phishing problem, we find important to visualize the life-cycle of the phishing attacks, and based on that categorize anti-phishing solutions. Based on our review of the literature, we depict a flowchart describing the life-cycle of phishing campaigns from the perspective of anti-phishing techniques, which is intended to be the most comprehensive phishing solutions flowchart. When a phishing campaign is started (e.g. by sending phishing emails to users), the first protection line is detecting the campaign.

The detection techniques are broad and could incorporate techniques used by service providers to detect the attacks, end-user client software classification, and user awareness programs. The ability to detect phishing campaigns can be enhanced whenever a phishing campaign is detected by learning from such experience.

For example, by learning from previous phishing campaigns, it is possible to enhance the detection of future phishing campaigns. Such learning can be performed by a human observer, or software (i.e. via a machine learning algorithm).

Once the phishing attack is detected, a number of actions could be applied against the campaign. According to our review of the literature, the following categories of approaches exist:

• Offensive defense — these approaches aim to attack phishing campaigns to render them less effective. This approach is particularly useful to protect users that have submitted their personal details to attackers.

• Correction — correction approaches mainly focus on taking down the phishing campaign. In case of phishingwebsites, this is achieved by suspending the hosting account or removing phishing files.

• Prevention — phishing prevention methods are defined differently in the literature depending on the context. In this survey, the context is attempting to prevent attackers from starting phishing campaigns in the future.

However, if the phishing campaign is not detected (let it be detected by a human or a software classifier), then none of these actions can be applied. This emphasizes on the importance of the detection phase.

•Detection Approaches

we consider any anti-phishing solution that aims to identify or classify phishing attacks as detection solutions. This includes:

• User training approaches — end-users can be educated to better understand the nature of phishing attacks, which ultimately leads them into correctly identifying phishing and non-phishing messages. This is contrary to the cate- gorization in where user training was considered a preventative approach. However, user training approaches aim at enhancing the ability of end-users to detect phishing attacks, and thus we categorize them under "detection". Further discussions on the human factor are

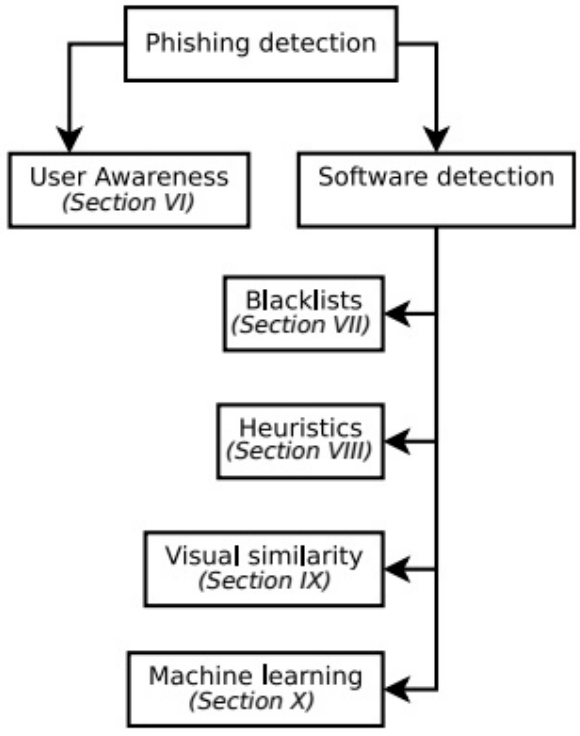

Fig. 3.3: An overview of phishing detection approaches

• Software classification approaches — these mitigation approaches aim at classifying phishing and legitimate messages on behalf of the user in an attempt to bridge the gap that is left due to the human error or ignorance. This is an important gap to bridge as user-training is more expensive than automated software classifiers, and user- training may not be feasible in some scenarios (such as when the user base is huge, e.g. PayPal, eBay, etc. . .)The performance of detection approaches can be enhanced during the learning phase of a classifier (whether the classifier is human or software).

In the case of end-users, their classification ability can be enhanced by improving their knowledge of phishing attacks by learning individually through their online experience, or by external training programs. In the case of software classifiers, this can be achieved during the learning phase of a Machine Learning-based classifier, or the enhancement of detection rules in a rule-based system.

Detection techniques not only help in directly protecting end-users from falling victims to phishing campaigns, but can also help in enhancing phishing honeypots 5 to isolate phishing spam from non-phishing spam. It is also important to note that the detection of phishing attackis the starting point of the mitigation of phishing attacks. As depicted in Figure 3, if a phishing campaign is not detected, none of the other mitigation approaches can beapplicable. For example, all of the mitigation techniques, such as correction, prevention and offensive defense depend on a functional and accurate detection phase.

•Offensive Defense Approaches

Offensive defense solutions aim to render phishing campaigns useless for the attackers by disrupting the phishing campaigns. This is often achieved by flooding phishing web-sites with fake credentials so that the attacker would have a difficult time to find the real credentials.

Two notable examples are:

1. A browser toolbar that submits fake information in HTML forms whenever a phishing websiteis encountered. According to BogusBiter, the detection of phishing websites is done by other tools. In other words, instead of simply showing a warning message to the end- user whenever a phishing website is visited, BogusBiter also submits fake data into HTML forms of the visited phishing website. Submitting fake data into the HTML forms is intended to disrupt the corresponding phishing campaigns, with the hope that such fake data may make

the attackers task of finding correct data (among the fake data) more difficult. This is an attempt to save the stolen credentials of other users that have been captured by the phishing campaign by contaminating the captured results with bogus data. However, the limitations are:

– Toolbars need to be installed on a wide enough user base to render this effective.

– If the user base is wide enough, BogusBiter may cause Denial of Service (DOS) floods against servers that host legitimate shared hosted websites as well, simply because one of the shared web-hosts may have a phishing content.

– Increased bandwidth demand.

– Non-standard HTML forms are not detected by BogusBiter.

– The empirical effectiveness of this solution is not accurately measured.

2. Similar to BogusBiter, except that BogusBiter relies on submissions from end-user clients,while Humboldt relies on distributed and dedicated clients over the Internet instead of end-user toolbars that may visit phishing sites, in addition to a mechanism to avoid causing DOS floods against servers. This can make Humboldt more effective against phishing websites due to the more frequent submission of data to phishing pages.

The limitations are:

– Increased bandwidth demand.

– Non-standard HTML forms are not detected by Humboldt.

– The empirical effectiveness of this solution is not accurately measured.

Although offensive defense approaches can theoretically make the attackers task more difficult in finding a victim's personal information, it is not known how difficult it really becomes. For example, a phisher might simply set up a script to test the credentials in a loop, and by using anonymous web surfing techniques attackers sessions will be difficult to track by the target web server. In other words, the actual returned security value of offensive defense approaches are not accurately evaluated and can be questioned.

• Correction Approaches

Once a phishing campaign is detected, the correction process can begin. In the case of phishing attacks, correction is the act of taking the phishing resources down. This is often achieved by reporting attacks to Service Providers. Phishing campaigns often rely on resources, such as:

• Websites — could be a shared web host owned by the phisher, a legitimate website with phishing content uploaded to it, or a number of infected end-user work-stations in a botnet 6 .

• E-mail messages — could be sent from a variety of sources, such as: free E-mail Service Provider (ESP) (e.g. Gmail, Hotmail, etc. . .), open Simple Mail Transfer Protocol (SMTP) relays or infected end-user machines that are part of a botnet.

• Social Networking services — web 2.0 services, such as Facebook and Twitter, can be used to deliver socially engineered messages to persuade victims to reveal their passwords

.• Public Switched Telephone Network (PSTN) and Voice over IP (VoIP) — similar to other forms of phishing attacks, attackers attempt to persuade victims to perform actions. However, the difference is that attackers attempt to exploit spoken dialogues in order to collect data (as opposed to clicking on links). Moreover, due to the way VoIP protocols (e.g. Session Initiation Protocol (SIP))function, and the way many VoIP provider systems are configured, spoofing Caller IDs are used by attackers as tools to increase their persuasion.

In order to correct such behavior, responsible parties (e.g. service providers) attempt to take the resources down. For example:

• Removal of phishing content from websites, or suspension of hosting services.

• Suspension of email accounts, SMTP relays, VoIP services

• Trace back and shutdown of botnets.

This also extends to the shutdown of firms that frequently provide services to phishing attackers.The shutdown process can be initiated by organizations that provide brand protection services to their clients, which may include banking and financial companies that are possible victims of phishing attacks. When phishing campaigns are identified, they can be reported to their hosting Internet and web hosting service providers for immediate shutdown.

Depending on the country where phishers and phishing campaigns exist, the penalties and procedures can differ.

• Prevention Approaches

The "prevention" of phishing attacks can be confusing, as it can mean different things depending on its context:

• Prevention of users from falling victim — in this case, phishing detection techniques will also be considered prevention techniques. However, this is not the context we refer to when "prevention" is mentioned in this survey.

• Prevention of attackers from starting phishing campaigns

— in this case, law suits and penalties against attackers by Law Enforcement Agencies (LEAs) are considered as prevention techniques.

Usually, LEA may take a number of weeks to complete their investigation and response procedures. Thus, it is common to apply prevention techniques after all other mitigationtechniques, which is due to the expensive nature of LEA investigations that makes them consume a relatively large period of time.

Once the sources of the phishing attacks are traced, LEA can then file law suits which in turn may issue penalties such as: imprisonment, fines and forfeiture of equipments used to convey the attacks.

3.3 EVALUATION METRICS

we find it useful to introduce the evaluation metrics used in the phishing literature. In any binary classification problem, where the goal is to detect phishing instances in a dataset with a mixture of phishing and legitimate instances, only four classification possibilities exist. See the confusion matrix presented in Table I for detailswhere N P →P is the number of phishing instances that are correctly classified as phishing, N L→P is the number of legitimate instances that are incorrectly classified as phishing, N P →L is the number of phishing instances that are incorrectly classified as legitimate, and N L→L is the number of legitimate instances that are correctly classified as legitimate.

	Classified as phishing	Classified as legitimate
Is phishing	$N_{P \to P}$	$N_{P \to L}$
Is legitimate	$N_{L \to P}$	$N_{L \to L}$

Classification Confusion Matrix

Based on our review of the literature, the following are the most commonly used evaluation metrics:

• True Positive (T P) rate — measures the rate of correctly detected phishing attacks in relation to all existing phishing attacks. See Equation (1) for details.

• False Positive (F P) rate — measures the rate of legitimate instances that are incorrectly detected as phishing attacks in relation to all existing legitimate instances. See Equation (2) for details.

• True Negative (T N) rate — measures the rate of correctly detected legitimate instances in relation to all existing legitimate instances. See Equation (3) for details.

• False Negative (F N) rate — measures the rate of phishing attacks that are incorrectly detected as legitimate in relation to all existing phishing attacks.See Equation (4) for details.

• Precision (P) — measures the rate of correctly detected phishing attacks in relation to all instances that were detected as phishing. See Equation (5) for details.

• Recall (R) — equivalent to TP. See Equation (6) for details.

• f 1 score — Is the harmonic mean between P and R. See Equation (7) for details.

• Accuracy (ACC) — measures the overall rate of correctly detected phishing and legitimate instances in relation to all instances. See Equation (8) for details.

) — measures the overall weighted rate of incorrectly detected phishing and legitimate instances in relation to all instances. See Equation (9) for details.

$$TP = \frac{N_{P\to P}}{N_{P\to P} + N_{P\to L}} \quad (1)$$

$$FP = \frac{N_{L\to P}}{N_{L\to L} + N_{L\to P}} \quad (2)$$

$$TN = \frac{N_{L\to L}}{N_{L\to L} + N_{L\to P}} \quad (3)$$

$$FN = \frac{N_{P\to L}}{N_{P\to P} + N_{P\to L}} \quad (4)$$

$$P = \frac{N_{P\to P}}{N_{L\to P} + N_{P\to P}} \quad (5)$$

$$R = TP \quad (6)$$

$$f_1 = \frac{2PR}{P+R} \quad (7)$$

$$ACC = \frac{N_{L\to L} + N_{P\to P}}{N_{L\to L} + N_{L\to P} + N_{P\to L} + N_{P\to P}} \quad (8)$$

$$W_{Err} = 1 - \frac{\lambda \cdot N_{L\to L} + N_{P\to P}}{\lambda \cdot N_{L\to L} + \lambda \cdot N_{L\to P} + N_{P\to L} + N_{P\to P}} \quad (9)$$

where λ weights the importance of legitimate instances as used previously. For example, if λ = 9, then W Err penalizes the misclassification of legitimate instances 9 times more than the misclassification of phishing instances.

CHAPTER 4

SPECIFICATION AND REQUIREMENTS

4.1 Functional Requirements

Functional requirements are associated with specific functions, tasks or behaviours the system must support. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements.A task-based functional requirements statement is a useful skeleton upon which to construct a complete requirements statement. Machine Learning techniques such as J48, Support Vector Machine (SVM), Logistic Regression (LR), Naive Bayes (NB) and Artificial Neural Network (ANN) were widely used to detect the phishing attacks. But getting a good quality training data is the biggest problems in the machine learning. So , a deep learning method called Deep Neural Network (DNN) is introduced to detect the phishing Uniform Resource Locators (URLs). That is the approach taken in this work. It can be helpful to think of non-functional requirements as adverbially related to tasks or functional requirements:

There are modules involved in this project:

- Initially, a feature extractor is used to construct a 30-dimension feature vector based on URL-based features, HTML-based features and domain based features.

- These features are given as input to the DNN classifier for Phishing attack detection.

- It consists of one input layer , multiple hidden layers and one output layer.

- The multiple hidden layers in DNN try to learn high-level features in an incremental manner.

- Finally , the DNN returns a probability value which represents the phishing URLs and Legitimate URLs.

- By using DNN the accuracy, precision and recall of phishing attack detection is improved.

4.2 Non-Functional Requirements

Non-functional requirements are requirements that specify criteria that can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that specify specific behavior or functions. In general, functional requirements define what a system is supposed to *do* whereas non-functional requirements define how a system is supposed to *be*. Non-functional requirements are often called qualities of a system. Qualities, aka. Non-functional requirements, can be divided into two main categories.

1. Execution qualities, such as security and usability, are observable at run time.
2. Evolution qualities, such as testability, maintainability, extensibility and scalability, are embodied in the static structure of the software system.

Scalability

The network-deployment cost for scaling up these systems must be manageable merely having the technology to provide a user service is not sufficient. The service-provider involvement requires that different infrastructure services be available. This information helps service providers to determine where to invest next. The data-collection facility is that service want to integrate into their service and system.

Interoperability

It is important that the interface is simple and intuitive Instead of making products and services ever more sophisticated, they must be made intuitive, simple, and useful in solving problems.

Reliability

It enhances the performance as well as the reliability of the system and communicate with more number of people at a time. It is imperative that the service reaches thousands of people, and that it is absolutely reliable.

Portability

In order to be more portable we use Application and even the is independent of the platform.

Extensibility

The application should be widely extensible.

Efficiency

The system should function in an efficient manner with proper acknowledgements and responses at high speed.

4.3 System Requirements Specifications

Software Requirements

The minimum requirements for detection and prevention of phishing attacks are:

- Operating System : Windows 7 and above
- Software : Anaconda
- Front End : Spyder (Python)
- Back End : Jupyter (Machine Learning , Deep Learning, Neural Network)

Hardware Requirements

The minimum hardware requirements are:

- Hard disk : 20 GB and above
- RAM : 4GB
- Processor : intel core i7
- Processor speed : 2.53 GHz and above

CHAPTER 5

METHODOLOGY

5.1 PROPOSED SYSTEM

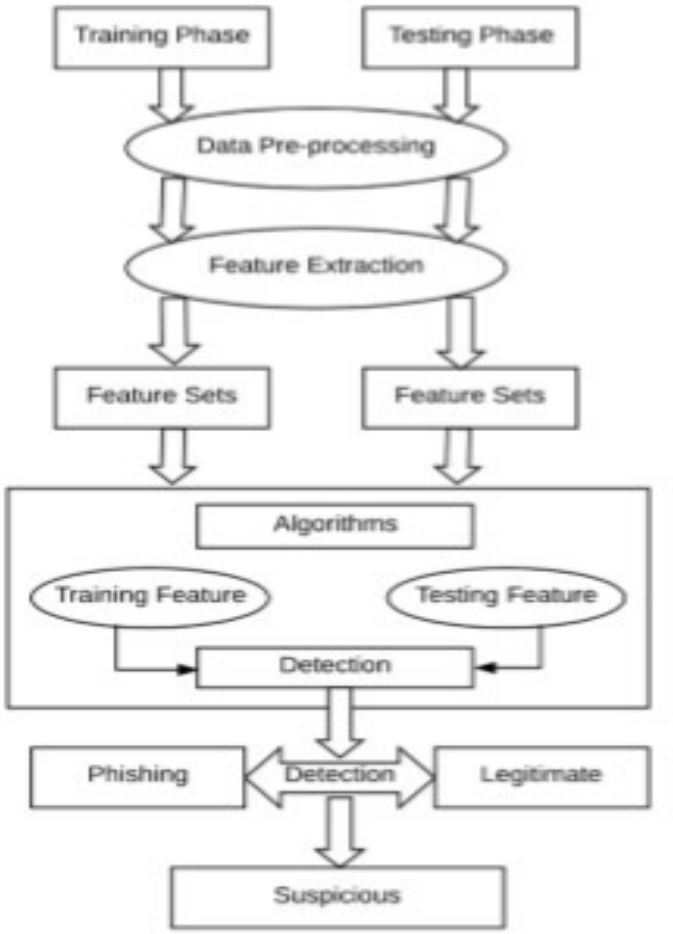

Fig 5.1: Architecture of the Proposed Work

The architecture of the proposed work consist of the data set consists of 2456 instances and 30 features. Value of attributes is in the form of integer -1, 0, and 1, -1 represents phishing, 0 denotes suspicious and 1 denotes legitimate First the data set has been processed to get mature data in desired format, then it is divided into two sections as training 70% and testing 30%.

Then the training and testing phases used to transform the raw data in a useful and efficient format or this data pre-processing phases “garbage-in”, ”garbage-out” is particularly applicable to data mining and machine learning projects.

Feature extraction transformation input data into a set of features. features are distinctive properties of input patterns that help in differentiating between the categories of input patterns. And feature extraction the data is divided into two sets of feature sets and then transported to algorithm section

In the algorithm phases the data is again moved to testing and training feature where leads to detection of the phishing website. Value of attributes is in the form of integer -1, 0, and 1, -1 represents phishing, 0 denotes suspicious and 1 denotes legitimate

PERFORMANCE METRIX

Confusion Matrix has been used to calculate different parameters such as accuracy, sensitivity or true positive rate (TPR), specificity or true negative rate (TNR), false positive (FP) and f-measure (Kahksha and Naaz, 2018).

- Accuracy is percentage of correct classification (true positive and negative) from overall numbers of instance.

  Accuracy (A) = (TP + TN) / (TP + TN + FP +FN)

- Sensitivity is percentage of correct positive classifications (true positive) from instances that are actually positive.

  Sensitivity (S)/Recall/ TPR = TP/ (TP+FN)

- Specificity is the percentage of positive records classified correctly out of all positive records

  Specificity (SS) = TN / (TN + FP)

- Precision is the percentage of the correct positive classification (true positive) from instances that predicate as positive.

  Precision = TP / (TP + FP)

- The F measure is defined as the weighted harmonic mean of the precision and recall of the test.

  F-Measure = 2 × Precision × Recall/ (Precision + Recall)

5.2 ALGORITHMS IMPLEMENTED

Data Collection

The dataset was gathered from MillerSmiles archive, PhishTank archive and Googleâ€™s searching operators. The data set consists of 2456 instances and 30 features. Value of attributes is in the form of integer -1, 0, and 1, -1 represents phishing, 0 denotes suspicious and 1 denotes legitimate First the data set has been processed to get mature data in desired format, then it is divided into two sections as training 70% and testing 30%. The experiments have been carried out using RStudio installed on windows 10. Four machine learning algorithms have been used in this work for detection of phishing websites. These include Decision tree, Random forest, Neural Network and Linear model.

Decision Tree

DT algorithms are machine learning algorithms (supervised learning) that can be used to solve classification and regression problems. In decision tree the data is subdivided into internal nodes and terminal nodes, internal node of a DT represents different attributes, the branches between the nodes represents the possible outcome that these attributes may have in the observed samples, whereas terminal nodes represent final result of the dependent variables (Zhao, 2012).

- Split the dataset which involves iterating over each row, cheacking if the attribute value is below or above the split value and assigning it to the left or right group
- Given a dataset, we must check every value on each attribute as a candidate split, evaluate the cost of split and find the best possible split we could make.
- Once the best split is found ,we can use it as a node in our decision tree
- Build a tree recursively until we get all the leaf nodes based on two criteria: Maximum tree depth and Minimum node record.
- Then we make predictions using the tree by navigating the tree upto its leaf node.
- Then we evaluate the accuracy of the algorithm using training test set and cross-validation set.

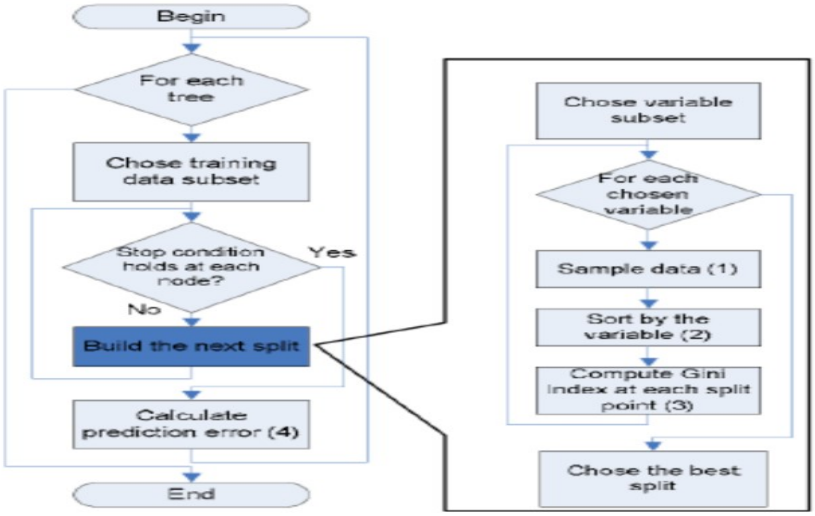

Fig 5.2: Diagrammatic representation of Decision Tree algorithm

Neural Network

Neural network builds a model that is based on the idea of multiple layers of neurons connected to each other, feeding the numeric data through the network, combining the numbers, to produce a final answer.

The two basic elements of evolutionary algorithms in Neural Network are:

- Variation operators (recombination and mutation)
- Selection process (selection of the fittest)

The common features of evolutionary algorithms are:

- Evolutionary algorithms are population-based.
- Evolutionary algorithms use recombination mix candidates of a population and create new candidates.
- On random selection evolutionary algorithm based.

Hence, on the basis of details and applied problems, we use various formats of evolutionary algorithms.
Some common evolutionary algorithms are:

- Genetic Algorithm Genetic Algorithm — It provides the solution for optimization problems. It provides the solution with the help of natural evolutionary processes. Like mutation, recombination, crossover, and inheritance.
- Genetic Programming — genetic programming provides a solution in the form of computer programs. By the ability to solve computational problems accuracy of a program measures.
- Evolutionary Programming — In a simulated environment to develop the AI we use it.
- Evolution Strategy It is an optimization algorithm. Grounded on the concepts of the adaptation and the evolution in biological science.
- Neuroevolution — To train neural networks we use Neuroevolution. By specifying structure and connection weights genomes uses to develop neural networks.

In all these Neural Network Algorithms, a genetic algorithm is the most common evolutionary algorithm.

Genetic Algorithm

Genetic algorithms, developed by John Holland's group from the early 1970s. It enables the most appropriate rules for the solution of a problem to be selected. So that they send their 'genetic material' (their variables and categories) to 'child' rules.

Here refer a like a set of categories of variables. For example, customers aged between 36 and 50, having financial assets of less than $20,000 and a monthly income of more than $2000.

A rule is the equal of a branch of a decision tree; it is also analogous to a gene. You can understand genes as units inside cells that control how living organisms inherit features of their parents. Thus, Genetic algorithms aim to reproduce the mechanisms of natural selection.

By selecting the rules best adapted to prediction and by crossing and mutating them until getting a predictive model.

Together with neural networks, they form the second type of algorithm. Which mimics natural mechanisms to explain phenomena that are not necessarily natural.
The steps for executing genetic algorithms are:

- Step 1: Random generation of initial rules — Generate the rules first with the constraint being that they must be all distinct. Each rule contains a random number of variables chosen by a user.
- Step 2: Selection of the best rules — Check the Rules in view of the aim by the fitness function to guide the evolution toward the best rules. Best rules maximize the fitness function and retain with the probability that increases as the rule improves. Some rules will disappear while others select several times.
- Step 3: Generation of new rules by mutation or crossing — First, go to step 2 until the execution of the algorithm stops. Chosen rules are randomly mutated or crossed. The mutation is the replacement of a variable or a category of an original rule with another.

A crossing of 2 rules is the exchange of some of their variables or categories to produce 2 new rules. A crossing is more common than mutation.
Neural Network Algorithms ends when 1 of the following 2 conditions meets:

- A specified number of iterations that reached.
- Starting from the generation of rank n, rules of generations n, n-1, and n-2 are (almost) identical

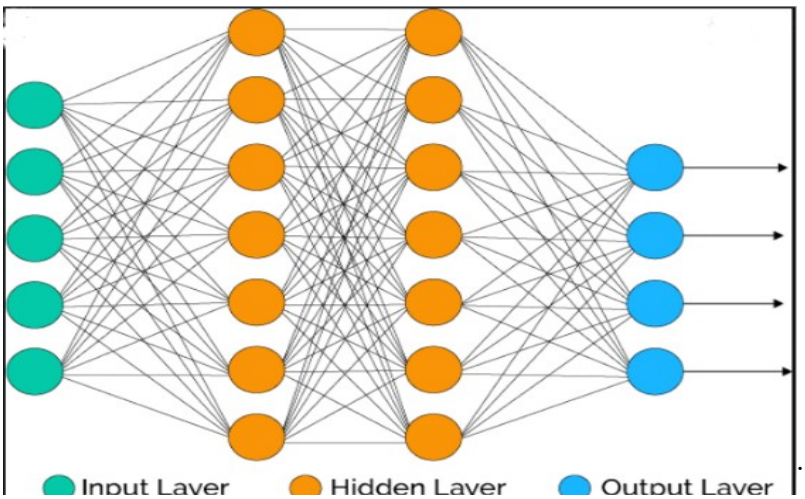

Fig 5.3: Diagrammatic representation of Decision Tree algorithm

Linear model

A linear regression model is the traditional method for fitting a statistical model to data. It is appropriate when the target variable is numeric and continuous. The family of generalized linear models extends traditional linear regression to targets with non-normal (non-gaussian) distributions. Linear regression models are iteratively fit to the data after transforming the target variable to a continuous numeric.

Random Forest

Random forest is an ensemble (i.e., a collection) of un-pruned decision trees. Ensemble models are often robust to variance and bias. Random forests are often used when we have large training datasets and particularly a very large number of input variables (hundreds or even thousands of input variables). The algorithm is efficient with respect to a large number of variables since it repeatedly subsets the variables available. A random forest model is typically made up of tens or hundreds of decision trees. In the RF approach, a large number of decision trees are created an error estimate is made for the cases which were not used while building the tree. That is called an OOB (Out-of-bag) error estimate which is mentioned as a percentage (Williams, 2009)

Algorithm for Construction of Random Forestis:

Step 1: Let the number of training cases be “n” and let the number of variables included in the classifier be “m”.

Step 2: Let the number of input variables used to make decision at the node of a tree be “p”. We assume that p is always less than “m”.

Step 3: Choose a training set for the decision tree by choosing k times with replacement from all “n” available training cases by taking a bootstrap sample. Bootstrapping computes for a given set of data the accuracy in terms of deviation from the mean data. It is usually used for hypothesis tests. Simple block bootstrap can be used when the data can be divided into non-overlapping blocks. But, moving block bootstrap is used when we divide the data into overlapping blocks where the portion “k” of overlap between first and second block is always equal to the “k” overlap between second and third overlap and so on We use the remaining cases to estimate the error of the tree. Bootstrapping is also used for estimating the properties of the given training data.

Step 4: For each node of the tree, randomly choose variables on which to search for the best split. New data can be predicted by considering the majority votes in the tree. Predict data which is not in the bootstrap sample. And compute the aggregate.

Step 5: Calculate the best split based on these chosen variables in the training set. Base the decision at that node using the best split.

Step 6: Each tree is fully grown and not pruned. Pruning is used to cut of the leaf nodes so that the tree can grow further.Here the tree is completely retained.

Step 7: The best split is one with the least error i.e. the least deviation from the observed data set.

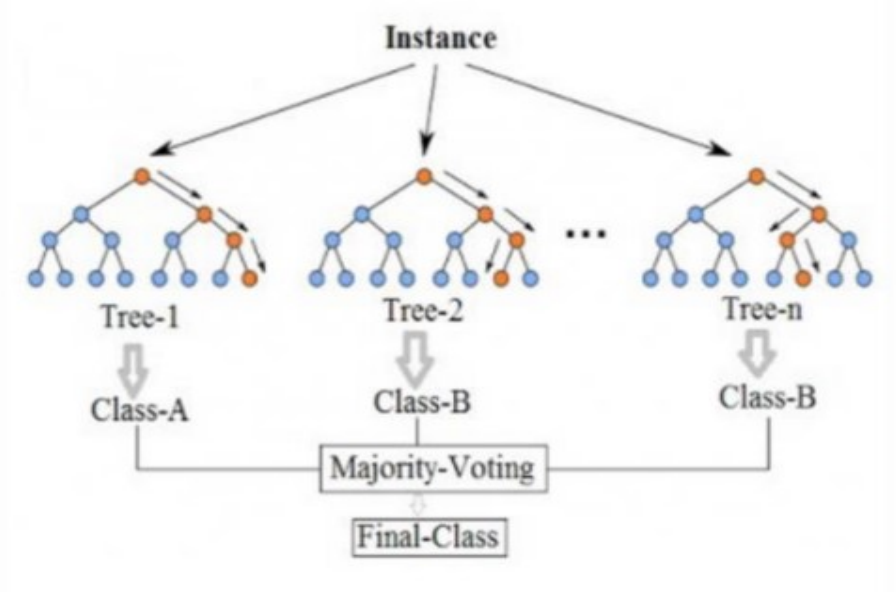

Fig 5.4: Diagrammatic representation of the working of a simple random forest

Technical approaches

A wide range of technical approaches are available to prevent phishing attacks reaching users or to prevent them from successfully capturing sensitive information.

- Filtering out phishing mail
- Browsers alerting users to fraudulent websites
- Augmenting password logins
- Monitoring and takedown
- Transaction verification and signing
- Multi Factor Authentication
- Email content redaction
- Limitations of technical responses

CHAPTER 6

IMPLEMENTATION

6.1 Characteristics Of Phishing Domains

Lets check the URL structure for the clear understanding of how attackers think when they create a phishing domain.

Uniform Resource Locator (URL) is created to address web pages. The figure below shows relevant parts in the structure of a typical URL.

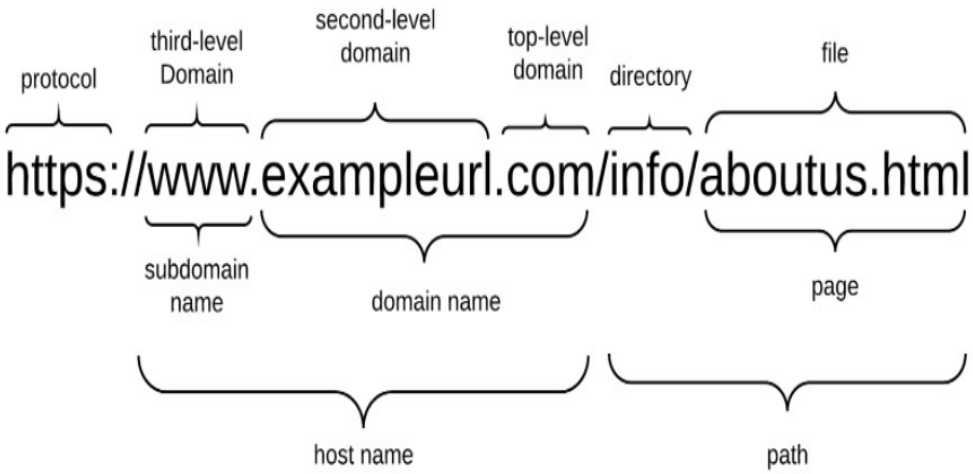

Fig 6.1: Structure of URL

It begins with a protocol used to access the page. The fully qualified domain name identifies the server who hosts the web page. It consists of a registered domain name (second-level domain) and suffix which we refer to as top-level domain (TLD). The domain name portion is constrained since it has to be registered with a domain name Registrar. A Host name consists of a subdomain name and a domain name. An phisher has full control over the subdomain portions and can set any value to it. The URL may also have a path and file

components which, too, can be changed by the phisher at will. The subdomain name and path are fullycontrollable by the phisher. We use the term FreeURL to refer to those parts of the URL in the rest of the article.

The attacker can register any domain name that has not been registered before. This part of URL can be set only once. The phisher can change FreeURL at any time to create a new URL. The reason security defenders struggle to detect phishing domains is because of the unique part of the website domain (the FreeURL). When a domain detected as a fraudulent, it is easy to prevent this domain before an user access to it.

Although the real domain name is active-userid.com, the attacker tried to make the domain look like paypal.com by adding Free URL. When users see paypal.com at the beginning of the URL, they can trust the site and connect it, then can share their sensitive information to the this fraudulent site. This is a frequently used method by attacker.

Features Used for Phishing Domain Detection

There are a lot of algorithms and a wide variety of data types for phishing detection in the academic literature and commercial products. A phishing URL and the corresponding page have several features which can be differentiated from a malicious URL. For example; an attacker can register long and confusing domain to hide the actual domain name **(**Cybersquatting, Typosquatting). In some cases attackers can use direct IP addresses instead of using the domain name. This type of event is out of our scope, but it can be used for the same purpose. Attackers can also use short domain names which are irrelevant to legitimate brand names and don't have any FreeUrl addition. But these type of web sites are also out of our scope, because they are more relevant to fraudulent domains instead of phishing domains.

Beside URL-Based Features, different kinds of features which are used in machine learning algorithms in the detection process of academic studies are used. Features collected from academic studies for the phishing domain detection with machine learning techniques are grouped as given below.

1. URL-Based Features

2. Domain-Based Features

3. Page-Based Features

4. Content-Based Features

URL-Based Features

URL is the first thing to analyse a website to decide whether it is a phishing or not. As we mentioned before, URLs of phishing domains have some distinctive points. Features which are related to these points are obtained when the URL is processed. Some of URL-Based Features are given below.

- Digit count in the URL
- Total length of URL
- Checking whether the URL is Typosquatted or not. (google.com → goggle.com)
- Checking whether it includes a legitimate brand name or not (apple-icloud-login.com)
- Number of subdomains in URL
- Is Top Level Domain (TLD) one of the commonly used one?

Domain-Based Features

The purpose of Phishing Domain Detection is detecting phishing domain names. Therefore, passive queries related to the domain name, which we want to classify as phishing or not, provide useful information to us. Some useful Domain-Based Features are given below

Page-Based Features

Page-Based Features are using information about pages which are calculated reputation ranking services. Some of these features give information about how much reliable a web site is. Some of Page-Based Features are given below.

- Global Pagerank
- Country Pagerank
- Position at the Alexa Top 1 Million Site

Some Page-Based Features give us information about user activity on target site. Some of these features are given below. Obtaining these types of features is not easy.

Content-Based Features

Obtaining these types of features requires active scan to target domain. Page contents are processed for us to detect whether target domain is used for phishing or not. Some processed information about pages are given below.

- Page Titles
- Meta Tags
- Hidden Text
- Text in the Body
- Images etc.

By analysingthese information, we can gather information such as;

- Is it required to login to website
- Website category
- Information about audience profile etc.

All of features explained above are useful for phishing domain detection. In some cases, it may not be useful to use some of these, so there are some limitations for using these features. For example, it may not be logical to use some of the features such as Content-Based Features for the developing fast detection mechanism which is able to analyze the number of domains between 100.000 and 200.000. Another example would be, if we want to analyze new registered domains Page-Based Features is not very useful. Therefore, the features that will be used by the detection mechanism depends on the purpose of the detection mechanism. Which features to use in the detection mechanism should be selected carefully.

Detection Process

Detecting Phishing Domains is a classification problem, so it means we need labeled data which has samples as phish domains and legitimate domains in the training phase. The dataset which will be used in the training phase is a very important point to build successful detection mechanism. We have to use samples whose classes are precisely known. So it means, the samples which are labeled as phishing must be absolutely detected as phishing. Likewise the samples which are labeled as legitimate must be absolutely detected as legitimate. Otherwise, the system will not work correctly if we use samples that we are not sure about.

Phishtank or other data resources, and create a new dataset to train our system with machine learning algorithms. The feature values should be selected according to our needs and purposes and should be calculated for every one of them.

There so many machine learning algorithms and each algorithm has its own working mechanism. In this article, we have explained ***Decision Tree Algorithm***, because I think, this algorithm is a simple and powerful one.

Initially, as we mentioned above, phishing domain is one of the classification problem. So, this means we need labeled instances to build detection mechanism. In this problem we have two classes: **(1)** phishing and **(2)** legitimate.

When we calculate the features that we've selected our needs and purposes, our dataset looks like in figure below. In our examples, we selected 12 features, and we calculated them. Thus we generated a dataset which will be used in training phase of machine learning algorithm.

A Decision Tree can be considered as an improved nested-if-else structure. Each features will be checked one by one. An example tree model is given below.

No.	1: domain	2: tld	3: brandName	4: editDbrandName	5: digitCount	6: length	7: isKnownTld	8: www	9: keywords	10: punnyCode	11: randomDomain	12:
	String	String	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	Numeric	
...	ayanasalon	com	0.0	1.0	0.0	10.0	0.0	0.0	0.0	0.0	0.0	
...	esteticabrasilbeauty	com	0.0	1.0	0.0	20.0	0.0	0.0	0.0	0.0	0.0	
...	erate365	com	0.0	1.0	3.0	8.0	0.0	0.0	0.0	0.0	0.0	
...	upstatescbusiness	com	1.0	1.0	0.0	17.0	0.0	0.0	1.0	0.0	0.0	
...	6-4c	com	0.0	0.0	2.0	4.0	0.0	0.0	0.0	0.0	0.0	
...	services-confirmatio...	com	1.0	1.0	0.0	23.0	0.0	0.0	1.0	0.0	0.0	
...	hmsinformatica	com	1.0	1.0	0.0	14.0	0.0	0.0	1.0	0.0	0.0	

Fig 6.2: Example for tree model

Generating a tree is the main structure of detection mechanism. Yellow and elliptical shaped ones represent features and these are called nodes. Green and angular ones represent classes and these are called leaves. The length is checked when an example arrives and then the other features are checked according to the result.

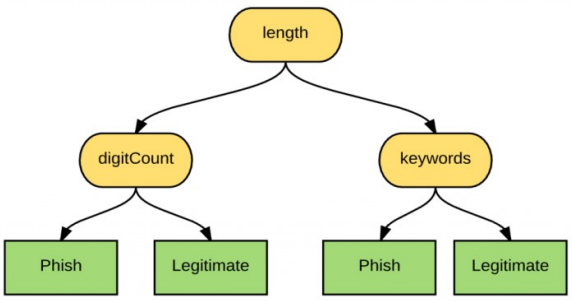

Fig 6.3: Decision tree

Now, the most important question about Decision Trees is not answered yet. The question is that which feature will be located as the root? which ones must come after the root? Choosing features intelligently effects efficiency and success rate of algorithms directly.So, how does decision tree algorithm select features?

Decision Tree uses a information gain measure which indicates how well a given feature separates the training examples according to their target classification. The name of the method is Information Gain. The mathematical equation of information gain method is given below.

$$Gain(S,A) = \underbrace{Entropy(S)}_{\text{original entropy of S}} - \underbrace{\sum_{v \in values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)}_{\text{relative entropy of S}}$$

High Gain score means that the feature has a high distinguishing ability. Because of this, the feature which has maximum gain score is selected as the root. **Entropy** is a statistical measure from information theory that characterizes (im-)purity of an arbitrary collection S of examples. The mathematical equation of Entropy is given below.

In the training phase, dataset is divided into two parts by comparing the feature values. In our example we have 14 samples. "+" sign representing phishing class, and "-" sign representing legitimate class. We divided these samples into two parts according to the *length* feature. Seven of them settle right, the other seven of them settle left. As shown in the figure below, right part of tree has high purity, so it means low Entropy Score (E), likewise left part of tree has low purity and high Entropy Score (E). All calculations were done according to the equations given above. Information Gain Score about the *length* feature is 0,151.

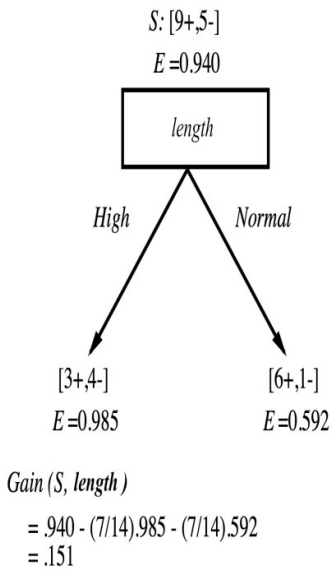

Fig 6.4: Decision Tree Algorithm calculates this information

The Decision Tree Algorithm calculates this information for every feature and selects features with maximum Gain scores. To growth the tree, leaves are changed as a node which represents a feature. As the tree grows downwards, all leaves will have high purity. When the tree is big enough, the training process is completed.

The Tree created by selecting the most distinguishing features represents model structure for our detection mechanism. Creating mechanism which has high success rate depends on training dataset. For the generalization of system success, the training set must be consisted of a wide variety of samples taken from a wide variety of data sources. Otherwise, our system may working with high success rate on our dataset, but it can not work successfully on real world data.

Spending time with the dataset

I have always loved to be a data-driven individual which (for me) means to use data wherever and whenever possible in a holistic way. Due to this fact, I generally spend a significant amount of time just by eyeballing at the data and see if I can find out any interesting fact

about it. Since I already had discovered the dataset, I decided to go ahead and load it up in an IPython *Notebook* (I use Jupyter Notebooks for this)

Here are the names of the features, the dataset has –

```
Index(['having_IP_Address', 'URL_Length', 'Shortining_Service',
       'having_At_Symbol', 'double_slash_redirecting', 'Prefix_Suffix',
       'having_Sub_Domain', 'SSLfinal_State', 'Domain_registeration_length',
       'Favicon', 'port', 'HTTPS_token', 'Request_URL', 'URL_of_Anchor',
       'Links_in_tags', 'SFH', 'Submitting_to_email', 'Abnormal_URL',
       'Redirect', 'on_mouseover', 'RightClick', 'popUpWidnow', 'Iframe',
       'age_of_domain', 'DNSRecord', 'web_traffic', 'Page_Rank',
       'Google_Index', 'Links_pointing_to_page', 'Statistical_report',
       'Result'],
      dtype='object')
```

Fig 6.5: Features of the Phishing Website Data Set

Features of the Phishing Websites Data Set

Let me discuss a few features from the above list –

- **having_At_Symbol**: Using "@" symbol in the URL leads the browser to ignore everything preceding the "@" symbol and the real address often follows the "@" symbol. Hence this can be used as a potential predictor in this problem

- **Favicon**: A favicon is a graphic image (icon) associated with a specific webpage. Many existing user agents such as graphical browsers and newsreaders show favicon as a visual reminder of the website identity in the address bar. If the favicon is loaded from a domain other than that shown in the address bar, then the webpage is likely to be considered a Phishing attempt. Hence, it is included as one of the predictors of the data.

- **Domain_registeration_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.

All the features *collectively* contribute to deciding if a website is vulnerable or not. Hence, I decided to not do any feature selection.

The dataset has a total of **30** predictors like the above and the label is saved in the feature named **Result**. A value of -**1** in the Result column denotes that the corresponding website is a phishing website and a value of **1** denotes that the corresponding website is a normal one. Each row (total **11055** rows) in the dataset represents a website by means of the quantified metrics (I mean the predictors). Here is the snap of the dataset containing the first ten entries -

	0	1	2	3	4	5	6	7	8	9
having_IP_Address	-1	1	1	1	1	-1	1	1	1	1
URL_Length	1	1	0	0	0	0	0	0	0	1
Shortining_Service	1	1	1	1	-1	-1	-1	1	-1	-1
having_At_Symbol	1	1	1	1	1	1	1	1	1	1
double_slash_redirecting	-1	1	1	1	1	-1	1	1	1	1
Prefix_Suffix	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
having_Sub_Domain	-1	0	-1	-1	1	1	-1	-1	1	-1
SSLfinal_State	-1	1	-1	-1	1	1	-1	-1	1	1
Domain_registeration_length	-1	-1	-1	1	-1	-1	1	1	-1	-1
Favicon	1	1	1	1	1	1	1	1	1	1
port	1	1	1	1	1	1	1	1	1	1
HTTPS_token	-1	-1	-1	-1	1	-1	1	-1	-1	1
Request_URL	1	1	1	-1	1	1	-1	-1	1	1
URL_of_Anchor	-1	0	0	0	0	0	-1	0	0	0
Links_in_tags	1	-1	-1	0	0	0	0	-1	1	1
SFH	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Submitting_to_email	-1	1	-1	1	1	-1	-1	1	1	1
Abnormal_URL	-1	1	-1	1	1	-1	-1	1	1	1
Redirect	0	0	0	0	0	0	0	0	0	0
on_mouseover	1	1	1	1	-1	1	1	1	1	1
RightClick	1	1	1	1	1	1	1	1	1	1
popUpWidnow	1	1	1	1	-1	1	1	1	1	1
Iframe	1	1	1	1	1	1	1	1	1	1
age_of_domain	-1	-1	1	-1	-1	1	1	-1	1	1
DNSRecord	-1	-1	-1	-1	-1	1	-1	-1	-1	-1
web_traffic	-1	0	1	1	0	1	-1	0	1	0
Page_Rank	-1	-1	-1	-1	-1	-1	-1	-1	1	-1
Google_Index	1	1	1	1	1	1	1	1	1	1
Links_pointing_to_page	1	1	0	-1	1	-1	0	0	0	0
Statistical_report	-1	1	-1	1	1	-1	-1	1	1	1
Result	-1	-1	-1	-1	1	1	-1	-1	1	-1

Fig 6.6: A sneak peek into the dataset

A sneak peek into the dataset

An interesting thing to observe from the snap is the values are -1, 0 or 1 (at least for the first ten entries). But is this true across the whole dataset?

Guess what? Yes, all the values in the dataset are either -1 or 0 or 1. **Why this information matters?**

Because it further raises a question — **do all the features represent *categories*?** From the above-mentioned document, it can be seen that yes, the way the features were generated, they are ought to be categories. But just going by a piece of text should not always be taken for granted. The facts and statements should always be verified with the data in hand. In this case, it is confirmed that *the features of the dataset are categorical in nature*.

The next question, that came to my mind —**what is the distribution of the classes in the dataset**?

This is again a very important question since it reveals if the dataset is an imbalanced dataset or not. And if it is sophisticated techniques like SMOTE, ADASYN etc can be incorporated to deal with the situation. I found out the following -

Fig 6.7: Distribution of the classes in the dataset

	having_IP_Address	URL_Length	Shortining_Service	having_At_Symbol	double_slash_redirecting	Prefix_Suffix	having_Sub_Domain	SSLfinal_State	Domain_
0	-1	1	1	1	-1	-1	-1	-1	
1	1	1	1	1	1	-1	0	1	
2	1	0	1	1	1	-1	-1	-1	
3	1	0	1	1	1	-1	-1	-1	
4	1	0	-1	1	1	-1	1	1	
5	-1	0	-1	1	-1	-1	1	1	
6	1	0	-1	1	1	-1	-1	-1	
7	1	0	1	1	1	-1	-1	-1	
8	1	0	-1	1	1	-1	1	1	
9	1	1	-1	1	1	-1	-1	1	

10 rows × 31 columns

Fig 6.8: Data Inspection

The balance of the classes is not very much poor it seems. It would be clearer once a baseline machine learning is built with the data and employed to make some predictions. With this, I ended the initial phase of data inspection.

Sweeping the dirt: Data cleaning

Ensuring the data is fed in a good shape to a machine learning is as important. Jargon as input = Jargon as output. The dataset does not suffer from missing value problem. Moreover, the feature values of dataset are in a uniform range. This is why feature scaling was also not required. All of the data was numeric in form, so that was an added advantage. The only issue that disturbed was representing the class of phishing websites as **-1**. In case of parameterized models like Logistic Regression, Neural Network and so on, (where a loss function is optimized to enhance their performance), the presence of negative label values turns out to be making the optimization process unstable. So, I decided to convert the **-1**values to **0**.

The process in the pipeline for me was to split the dataset into training and test sets. This is specifically very important in this problem. Let's move to the next section to discover why.

(Strategic) Splits are just as important

Sometimes, for a particular dataset, separate train and test splits are provided with it. Sometimes, even a validation set too. But when such splits are not available, you need to take the responsibility of carefully determining the splits. Data splitting has a direct relation to the **kind of data** you are dealing with. For example, if you are creating a machine learning model to predict on time-series data or any data that has some temporal characteristics to it, a good validation split should contain future data whereas the model should be trained with the past observations. In this problem, the data does not have any kind of temporal relationship. So, I created the splits by first randomly shuffling rows of the dataset and then split the shuffled samples in an **80:20** ratio (train:test). However, I did create two more splits from the same dataset where it was not done in a randomized fashion, rather a test set was extracted with respect to a specific row index (maintaining the same split ratio). To know more on the nitty-gritty details of an effective train/validation/test split, readers are encouraged to read this article by Rachel Thomas.

My favorite piece of the cake: Machine Learning

Finally, I was ready to begin my machine learning modeling experiments. From looking very closely at the values of the dataset, I had a gut feeling of applying linear classifiers to it and I followed my trust. I started with an off-the-shelf Logistic Regression model with the default hyperparameters settings provided by **scikit-learn**. I got a test accuracy of **93.71%**. Not a bad baseline at all. I investigated the model further and extracted other performance metrics like **precision**, **recall** and **f1-score**. Here's the classification report:

	precision	recall	f1-score	support
Phishing Websites	0.94	0.92	0.93	974
Normal Websites	0.94	0.95	0.94	1237
avg / total	0.94	0.94	0.94	2211

Fig 6.9: Classification report from the baseline model

For a dead simple model like this, the above scores totally make sense. On average it is able to bring out the true positives and true negatives quite effectively (given its simplicity). But can this be improved further? Albeit, yes!

Pushing the base model's predictive performance

There were many things on the plate for me to do for pushing the evaluation metrics of the model. I chose the most obvious one — hyperparameter tuning. I used randomized searching to find an optimal set of hyperparameters for the model to train with. For the hyperparameter grid, I chose **penalty**, **C**, **tol**and **max_iter** as the hyperparameters. Check out the scikit-learn documentation in order to know more about these. But to my wonder, the default values worked the best for the model on this dataset. This fact further left me in the awe of re-believing that there is no free-lunch theorem in machine learning. So, what's next? Yes, you guessed it right — **neural networks**!

A bit of deep learning into the picture

To kickstart the neural network modeling, I decided to use the model configurations from the above-mentioned paper (Swarm Intelligence Approaches for Parameter Setting of Deep

Learning Neural Network: Case Study on Phishing Websites Classification) I mentioned above. Here's how the model's topology looks like -

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_1 (Dense)              (None, 40)                1240
_________________________________________________________________
dense_2 (Dense)              (None, 30)                1230
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 31
=================================================================
Total params: 2,501
Trainable params: 2,501
Non-trainable params: 0
```

Fig 6.10: The topology of the initial network

Following are the model's hyperparameters' settings with which the network was trained:

- Optimizer: Adam
- Learning rate: 0.0017470
- Batch size: 10
- Number of epochs: 100
- Loss function: Binary cross-entropy

And guess what? The scores bumped up (but little).The model yielded an accuracy of **96.52%** and generated significantly better results even on the other metrics -

	precision	recall	f1-score	support
Phishing Websites	0.95	0.97	0.96	974
Normal Websites	0.98	0.96	0.97	1237
avg / total	0.97	0.97	0.97	2211

Fig 6.11: Classification report from the initial neural network

At this point, I was pretty satisfied with the predictive performance of the network. But then *modern deep learning* happened.

Going beyond with modern deep learning practices

A practitioner will always try to use the available resource to find the best possible solution within a feasible amount of time. So did I. I decided to use the deep learning library **fastai** to take advantage of its modern deep learning practices. There is a misconception in the machine learning community that tabular datasets (like the one we have here in this project) are not well suited for the modern deep learning practices. The **fastai** library tends to break this quite efficiently

The improvements in the results were not that hard to speculate. An accuracy score of **97.02%**. Here's the confusion matrix -

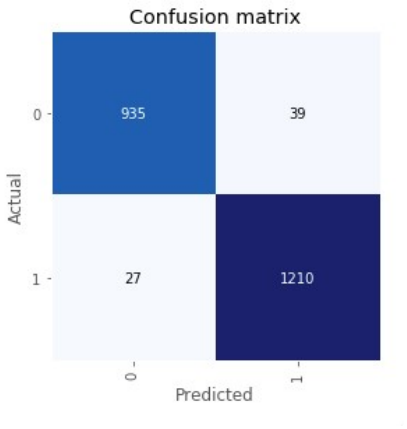

Fig 6.12: Confusion matrix of the final model

Confusion matrix of the final model

For this particular experiment, I decided to use the other split I had prepared and as a result, the performance of the model was similar to the previous models. It further confirms that due to this splitting strategy the model might have missed out on the important training examples that were there because of the random shuffling. I even validated this with a few other randomly shuffled splits and the model on an average generated the same result. So, this confirms that there is no temporal pattern in the data and a sense of randomization is needed for better predictive performance.

CHAPTER 7

RESULTS AND DISCUSSIONS

Fig 7.1: Home page

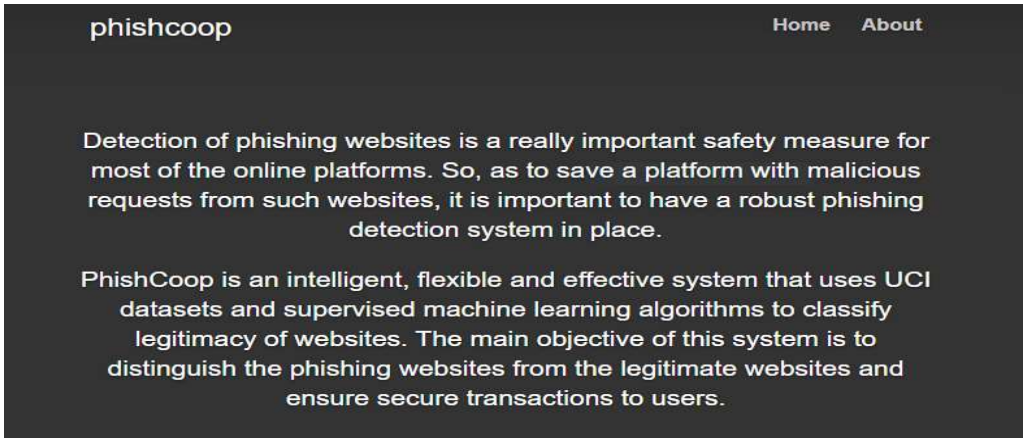

Fig 7.2: About page

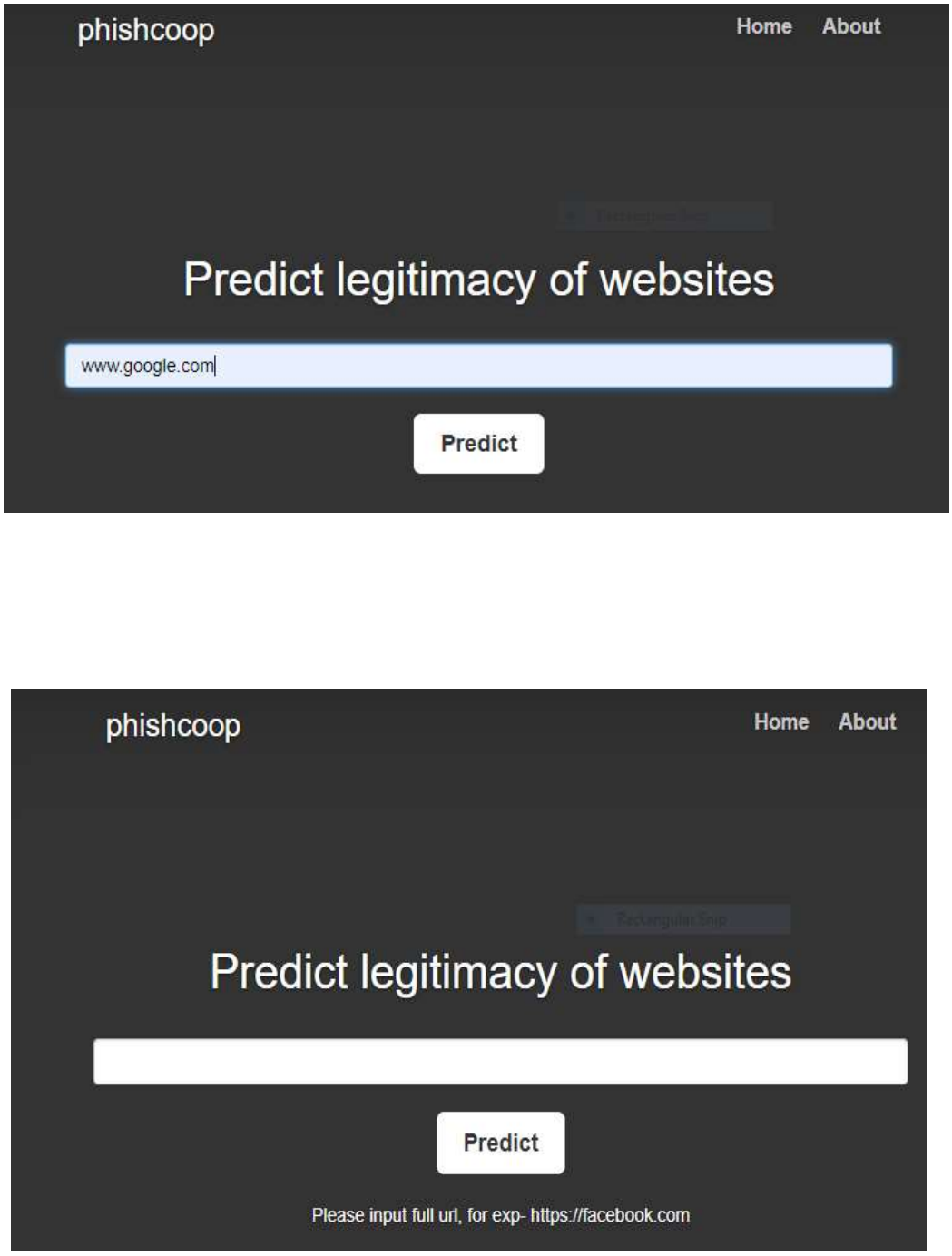

Fig 7.3: Example for predicting legitimacy of websites

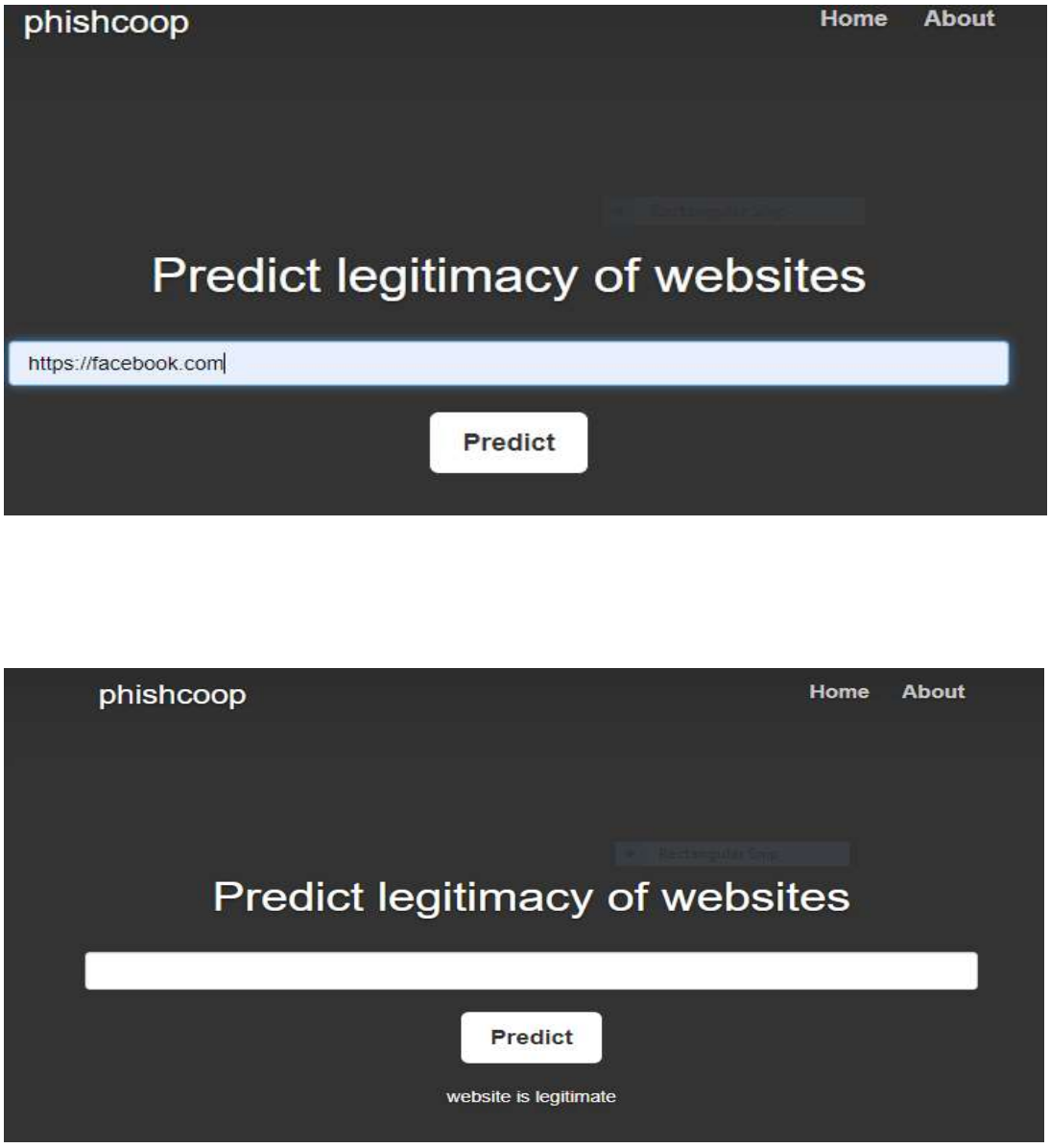

Fig 7.4: Example for website is legtimate

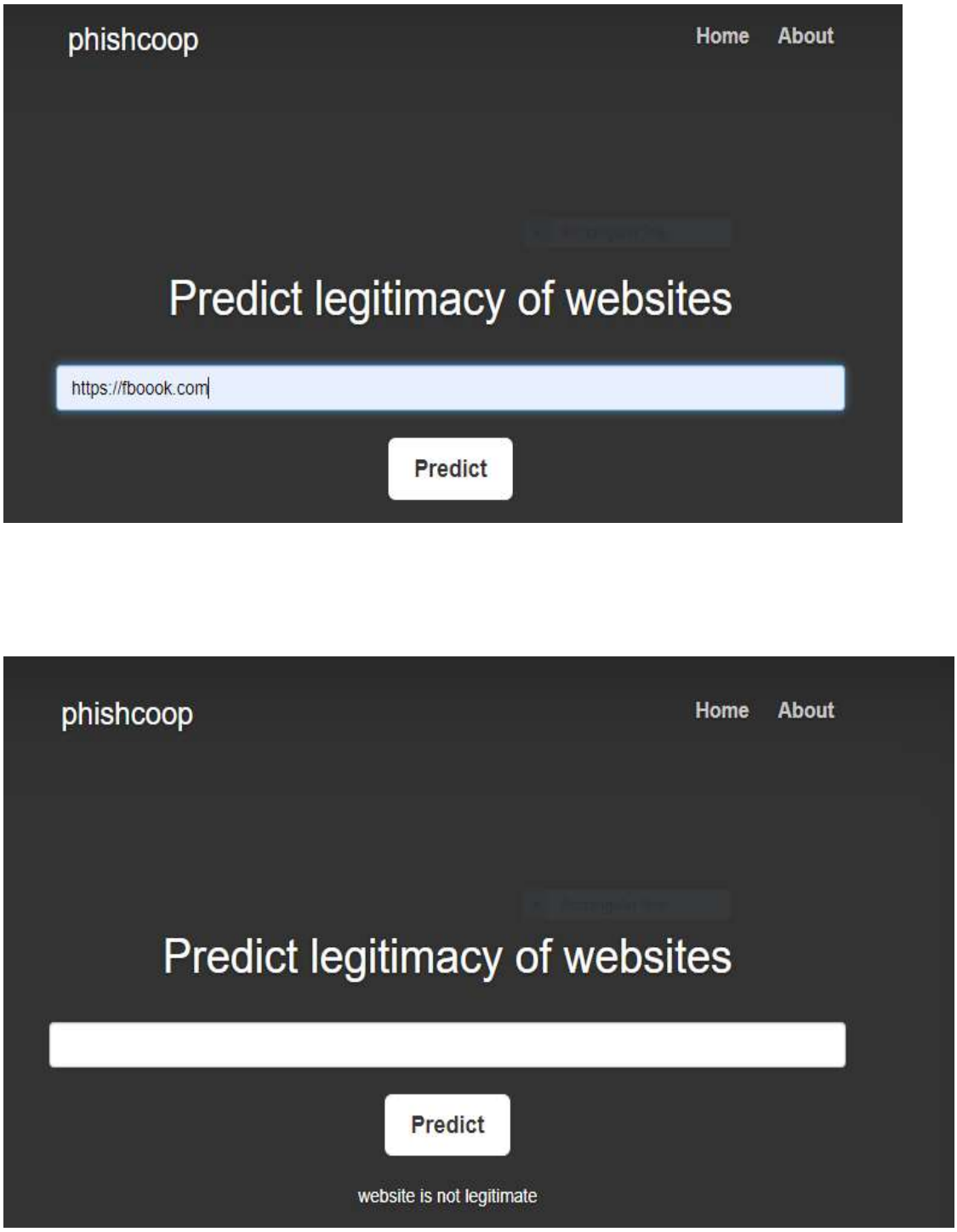

Fig 7.5: Example for website is not legitimate

In this work 2456 websites having 30 attributes have been investigated from the data taken from UCI Machine learning repository data. Rstudio has been used here for the implementation. The decision tree, random forest, neural network and linear model has been implemented on the above dataset and the results in terms of accuracy, true positive rate (TPR), true negative rate (TNR), Precision, F measure and false positive rate. These values have been calculated using formulas given in performance metrices and using confusion matrix on testing data set.

	DT	RF	NN	LM
Accuracy	90.4%	95.7%	90.70%	92.10%
TPR	93.2%	96.1%	84.00%	93.80%
TNR	88.7%	95.2%	97.90%	90.00%
Precision	83.2%	93.7%	94.00%	92.00%
F-Measure	87%	94%	90.40%	92.80%
Error Rate	9.5%	4.3%	9.2%	8.0%

Fig 7.6: Performance of different Algorithms

Table 1 above shows a comparison of algorithms applied for the phishing websites dataset in the testing phase implemented on R software. In random forest model error rate has been plotted against different number of trees, namely 100, 200, 300, 400, and 500. In can be seen from the results that Random Forest with 500 trees has the minimum error rate of 0.043. Subsequently, the random forest setup has been used for evaluation and comparison with other methods

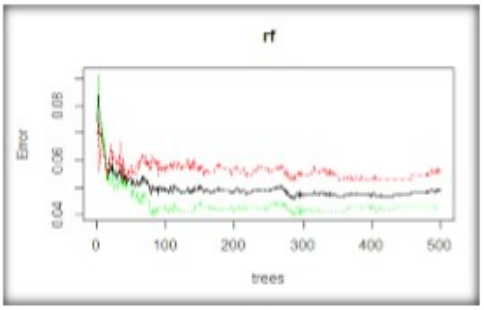

Fig 7.7: Error rate for different number of trees in random forest

Error rate of different machine learning algorithms has been calculated using confusion matrix and random forest has been found to have minimum error rate of 4.3%, linear model showed error rate of 8% followed by neural network with 9.2% whereas decision tree has been found to perform worst with error rate of 9.5% as shown in Fig. 3.

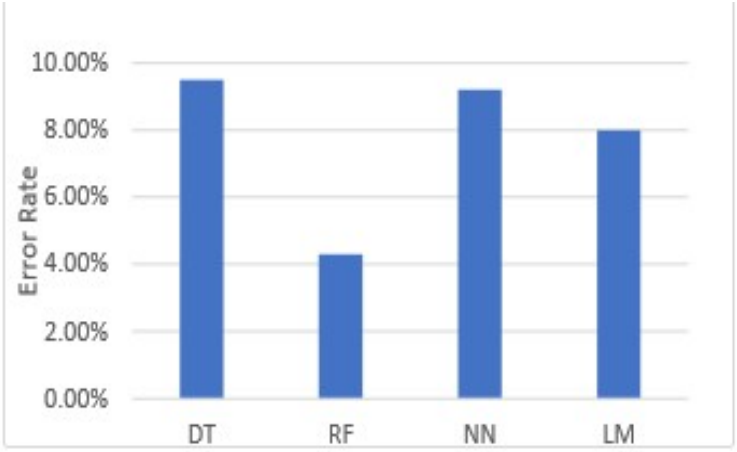

Fig 7.8: Comparison of error rate

In Phishing website classification, false positives (FP) are legitimate websites that are misclassified as phishing and false negatives (FN) are phishing websites that are misclassified as legitimate.

	DT	RF	NN	LM
FPR	11.2%	4.7%	0.02%	0.09%
FNR	0.067%	0.038%	0.015%	0.06%

Fig 7.9: False Positive rate (FPR), and False Negative rate (FNR)

In the end all the above algorithms have been compared in terms of accuracy precision, recall, and F-measure.

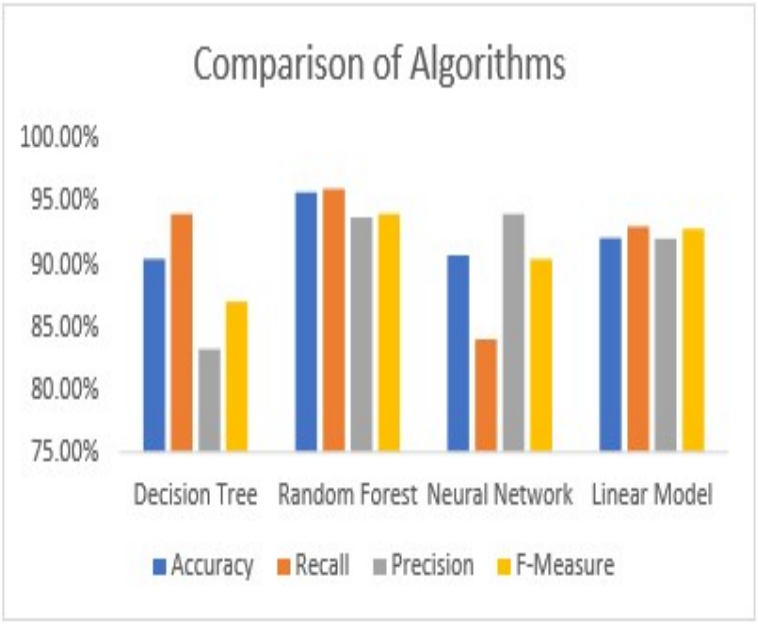

Fig. 7.10: Comparison of Algorithms

As shown in Fig. 4 the Random Forest algorithm performs better as compared to other algorithms in term of different parameters. It achieved highest accuracy of all at 95.70% whereas other algorithms perform with accuracy of 90.4% (DT), 90.7% (NN) and 92.1% (LM). ROC of the phishing dataset has been created for RF, NN and DT by plotting specificity verses sensitivity as shown in the Fig. 5.

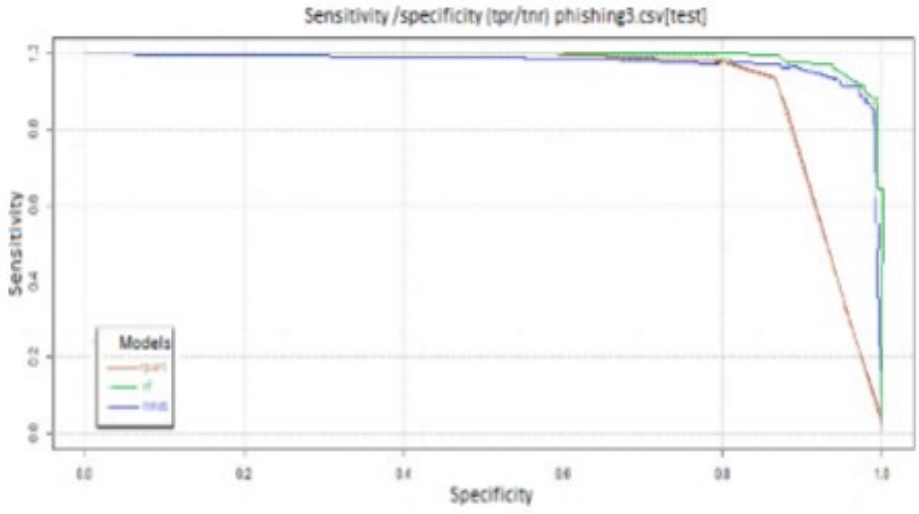

Fig 7.11: ROC of different algorithms

From the ROC curve RF is found better as compared to other algorithms.

Precision verses recall plot is shown in the Fig. 6.

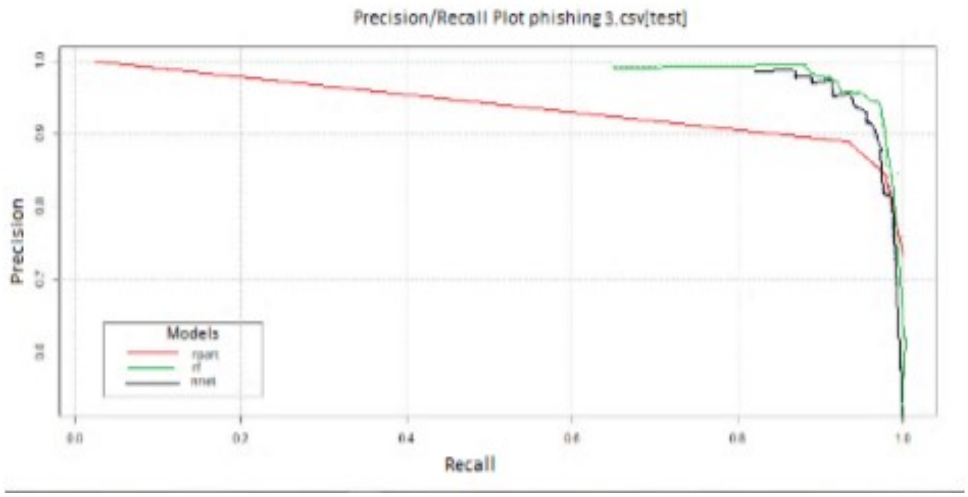

Fig 7.12: Precision verses recall

Here are a few steps a company can take to protect itself against phishing:

- Educate your employees and conduct training sessions with mock phishing scenarios.
- Deploy a SPAM filter that detects viruses, blank senders, etc.
- Keep all systems current with the latest security patches and updates.
- Install an antivirus solution, schedule signature updates, and monitor the antivirus status on all equipment.
- Develop a security policy that includes but isn't limited to password expiration and complexity.
- Deploy a web filter to block malicious websites.
- Encrypt all sensitive company information.
- Convert HTML email into text only email messages or disable HTML email messages.
- Require encryption for employees that are telecommuting.

CONCLUSION

Phishing is growing continuously irrespective of intelligence security development, there is definitely need of special care toward safeguarding of people being cheated. In this work different machine learning algorithms have been compared on phishing dataset and found that random forest works better in terms of accuracy, error rate and other parameters. The proposed algorithm has also been compared with existing similar works and it has been found that the purposed model achieves considerably higher accuracy as compared to works reported by different authors.

REFRENCES

[1] Hodžić, A., and Kevrić, J, Comparison of Machine Learning Techniques in Phishing Website Classification. International Conference on Economic and Social Studies.

[2] Shrivas, A. K., & Suryawanshi, R, Decision Tree Classifier for Classification of Phishing Website with Info Gain Feature.

[3] Mohammad, R. M., Thabtah, F.,& McCluskey, L,"Intelligent rule-based phishing websites classification".

[4] Kalaiselvan, O. & Edwinraja, S, "Predicting Phishing Websites using Rule Based Techniques".

[5] Chen, H., Vasardani, M., & Winter, S, Geo-referencing Place from Everyday Natural Language Descriptions.

[6] M. Zouina and B. Outtaj, "A novel lightweight URL phishing detection system using SVM and similarity index".

[7] E. Buber, Ö. Demir and O. K. Sahingoz, "Feature selections for the machine learning based detection of phishing websites".

[8] S.Marchal, K. Saari, N. Singh and N. Asokan, "Know your phish: Novel techniques for detecting phishing sites and their targets".

[9] Bhagyashree E. Sananse, Tanuja K .Sarode: Phishing URL Detection: A Machine Learning and Web Mining-based Approach.

[10] MonaliDeshmukh1 , Shraddha K. Popat2 UG Student1 , Assistant Professor2 Department of Computer Engineering D .Y. Patil College of Engineering, Pune, India.