Chapter 1

PREAMBLE

Web service is a communication protocol and software between two electronic devices over the Internet. Web services extends the World Wide web infrastructure to provide the methods for an electronic device to connect to other electronic devices. Web services are built on top of open communication protocols such as TCP/IP, HTTP, Java, HTML, and XML. Web service is one of the greatest inventions of mankind so far, and it is also the most profound manifestation of computer influence on human beings

With the rapid development of the Internet and the increasing popularity of electronic payment in web service, Internet fraud and web security have gradually been the main concern of the public. Web Phishing is a way of such fraud, which uses social engineering technique through short messages, emails, and WeChat to induce users to visit fake websites to get sensitive information like their private account, token for payment, credit card information, and so on.

1.1 Introduction

In this cyber world, most of the people communicate with each other either through a computer or a digital device connected over the Internet. The number of people using e-banking, online shopping and other online services has been increasing due to the availability of convenience, comfort, and assistance. An attacker takes this situation as an opportunity to gain money or fame and steals sensitive information needed to access the online service websites. Phishing is one of the ways to steal sensitive information from the users. It is carried out with a mimicked page of a legitimate site, directing online user into providing sensitive information. The term phishing is derived from the concept of ‘fishing’ for victims’ sensitive information. The attacker sends a bait as mimicked webpage and waits for the outcome of sensitive information. The replacement of ‘f’ with ‘ph’ phoneme

is influenced from phone phreaking, a common technique to unlawfully explore telephone systems. The attacker is successful when he makes a victim to trust the fake page and gains his/her credentials related to that mimicked legitimate site. Anti-Phishing Working Group (APWG) is a non-profit organization which examines phishing attacks reported by its member companies such as iThreat Cyber Group, Internet Identity (IID), MarkMonitor, Panda Security and Forcepoint. It analyses the attacks and publishes the reports periodically. It also provides statistical information of malicious domains and phishing attacks taking place in the world.

Online users fall for phishing due to various factors such as:

1. Inadequate knowledge of computer systems.

2. Inadequate knowledge on security and security indicators. (In the current scenario, even the indicators are being spoofed by the phishers.)

3. Inadequate attention to warnings and proceeding further by undermining the strength of existing tools. (abnormal behaviour of toolbars)

4. Inadequate attention to the visual deceptive text in URL and Website content.

Phishing attacks take place through various forms such as email, websites and malware. To perform email phishing, attackers design fake emails which claim to be arriving from a trusted company. They send fake emails to millions of online users assuming that at least thousands of legitimate users would fall for it.

Phishing is the process whereby someone attempts to obtain your confidential information, such as your passwords, your credit card number, your bank account details or other information protected by the Data Protection Act. Such attempts, often referred to as Phishing attacks, are usually primitive and obvious; however, please be aware that they are becoming more sophisticated.

1.2 Existing System

1.2.1 Heuristic-based Techniques

These techniques use features extracted from the phishing website to detect phishing attack. Some of the phishing sites do not have common features resulting in poor detection rate using this mechanism. As this approach does not use list-based comparison, it results in less false positives and less false negatives. This technique detects zero-day phishing attacks which the list-based techniques fail to detect.

Disadvantages

- It has less accuracy compared to list-based techniques as there is no guarantee of existence of these features in all phishing websites.
- An attacker can bypass the heuristic features once he knows the algorithm or features used in detecting phishing sites thereby reaches his goal of stealing sensitive information.

1.2.2 Visual Similarity-based Approach

The main objective of the phisher is to deceive the user by designing an exact image of legitimate site such that the user does not get any suspicion on the phishing site. Hence, the anti-phishing techniques compare suspicious website image with legitimate image database to get the similarity ratio, used for the classification of suspicious websites. The website is classified as phishing when the similarity score is greater than a certain threshold else it is treated as legitimate.

Disadvantages

- Image comparison of suspicious website with entire legitimate database store takes more time complexity.
- More space to store legitimate image database.
- Web page with animated website compared with phishing website leads to the low percentage of similarity that leads to high false negative rate. This technique fails,

when the background of web page is slightly changed without deviating from visual appearance of legitimate site.

1.2.3 Machine learning-based Techniques

Nowadays, most of the researchers are concentrating on the use of machine learning algorithms (ML) applied on the features extracted from the websites to detect phishing attacks. These techniques are a combination of heuristic methods and machine learning algorithms, i.e., dataset used by the machine learning algorithms is extracted through heuristic methods. Some of the machine learning algorithms are sequential minimum optimization (SMO), J48 tree, Random Forest (RF), logistic regression (LR), multilayer perceptron (MLP), Bayesian network (BN), support vector machine (SVM) and AdaBoostM1 etc. As ML-based techniques are based on heuristic features, they are able to identify the zero-day phishing attacks which make them advantageous than list-based techniques.

Disadvantages

- These techniques work efficiently on the large sets of data.

1.3 Problem Statement

In website phishing, attacker builds a website which looks like a replica of legitimate site and draws the online user to the website either through advertisements in other websites or social networks such as Facebook and Twitter etc. Some of the attackers are able to manage phishing websites along with security indicators such as green padlock, HTTPS connection etc. Hence, HTTPS connection is no longer guaranteed to decide legitimacy of a website. This problem to be effectively handled through implementing an efficient phishing detection system.

1.4 Objective of the Project

Phishing attacks are one of the most common and least defended security threats today. Objective of study to identify phishing attacks using five machine learning algorithms.

The proposed system handles feature selection through learning algorithm, after feature selection, training and prediction is done. The objective of our study to find an efficient algorithm, which achieves highest accuracy and an extension is added to the Google Chrome web browser such that, if any phishing website is visited, then an alert message is displayed on that web page.

In response to this increase in phishing attacks, phishing detection techniques have been the focus of considerable research. Typical phishing detection techniques include the blacklist-based detection method and the heuristic-based technique. The blacklist-based technique maintains a uniform resource locator (URL) list of sites that are classified as phishing sites; if a page requested by a user is present in that list, the connection is blocked. This technique is commonly used and has a low false-positive rate; however, its accuracy is determined by the quality of the list that is maintained. Consequently, it has the disadvantage of being unable to detect temporary phishing sites. The heuristic-based detection technique analyses and extracts phishing site features and detects phishing sites using that information. In our project, we propose a new heuristic-based phishing technique that resolves the limitation of the blacklist-based technique. The proposed technique extracts the features in URLs of user-requested pages and applies those features to determine whether a requested site is a phishing site. This technique can detect phishing sites that cannot be detected by blacklist-based techniques; therefore, it can help reduce damage caused by phishing attacks. This problem is to be effectively handled through implementing an efficient phishing detection system.

1.5 Proposed System

- ❖ Add new heuristic features with machine learning algorithms to reduce the false positives in detecting new phishing sites.

- Made an attempt to identify the best machine learning algorithm to detect phishing sites with high accuracy than the existing techniques.
- Used five machine learning algorithms (Logistic regression (LR), K-Nearest Neighbour (KNN), Random Forest (RF), support vector machine (SVM) and Decision Tree) to classify the websites as legitimate and phishing.
- Based on the experimental observations, Random Forest outperformed the others.
- The choice of considering these machine learning algorithms is based on the classifiers used in the recent literature.
- A google chrome extension is designed using the Random Forest algorithm to raise an alert when a phishing website is visited.

Chapter 2

LITERATURE SURVEY

A literature survey depicts the various analysis and research made in the field of interest of the project and the results already published. It is an important part as it gives a direction in the area of your research. It helps to set a goal for your analysis- thus deriving at the problem statement. After taking into account various parameters and the extent of the project, the following papers were analysed:

A. Pratik Patil, Prof. P.R. Devale: "Protecting user against phishing using Anti-Phishing" -

1) Anti-Phishing is used to avoid users from using fraudulent web sites which in turn may lead to phishing attack. Here, Anti-Phishing traces the sensitive information to be filled by the user and alerts the user whenever he/she is attempting to share his/her information to an untrusted web site. The much effective elucidation for this is cultivating the users to approach only for trusted websites.
2) However, this approach is unrealistic. Anyhow, the user may get tricked. Hence, it becomes mandatory for the associates to present such explanations to overcome the problem of phishing. Widely accepted alternatives are based on the creepy websites for the identification of "clones" and maintenance of records of phishing websites which are in hit list.
3) String Kernels: In contrast of distance-based kernels, string kernels define the similarity between pair of documents by measuring the total occurrence of shared substrings of length k in feature space F. In this case, the kernel is defined via an explicit feature map. In our experiments we adopted two classes of string kernels: the position-aware string kernel which takes advantage of positional information of

characters/substrings in their parent strings and the position-unaware string kernel which does not. We applied Weighted Degree kernel (WD) and Weighted Degree kernel with Shift (WDs) for position-aware kernels. Additionally, for position-unaware kernels, Subsequence String kernel (SSK), Spectrum kernel and Inexact String Kernels such as Mismatch kernel, Wildcard kernel and Gappy kernel.

B. Abdul Ali Ahmed: "Real Time Detection Of Phishing Websites" -

Web Spoofing lures the user to interact with the fake websites rather than the real ones. The main objective of this attack is to steal the sensitive information from the users. The attacker creates a 'shadow' website that looks like the legitimate website. This fraudulent act allows the attacker to observe and modify any information from the user. This paper proposes a detection technique of phishing websites based on checking Uniform Resources Locators (URLs) of web pages. The proposed solution can distinguish between the legitimate web page and fake web page by checking the Uniform Resources Locators (URLs) of suspected web pages. URLs are inspected based on characteristics to check the phishing web pages. The detected attacks are reported for prevention. The performance of the proposed solution is evaluated using Phish tank and Yahoo directory datasets. The obtained results show that the detection mechanism is deployable and capable to detect various types of phishing attacks maintaining a low rate of false alarms.

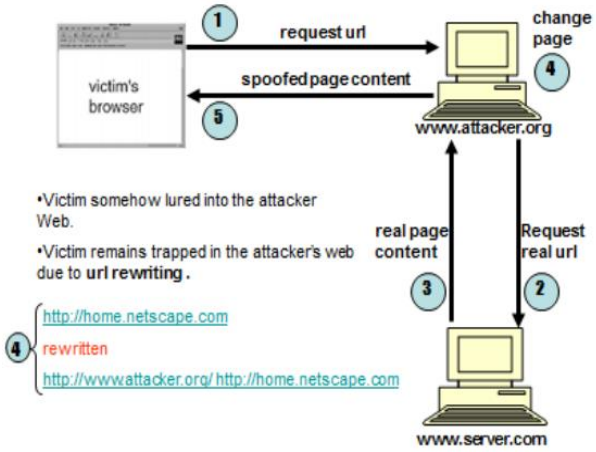


Fig. 2.1 Real-time detection of phishing websites

There are many researches conducted to detect web spoofing attacks. However, these researches are not effective enough to stop the sophisticated attack of web spoofing. The use of various media communication such as social network leads to the increase of the numbers of attacks. According to [4], 70% of successful phishing attacks are launched through social network. In fact, the lack of awareness and education on web spoofing attack causes the fall of the victims. Inability to distinguish between the fake and legitimate web pages is still a challenge in the existing prevention solutions of web spoofing. Moreover, the current solutions of antivirus, firewall and designated software do not fully prevent the web spoofing attack. The implementation of Secure Socket 978 Layer (SSL) and digital certificate (CA) also does not protect the web user against such attack. In web spoofing attack, the attacker diverts the request to fake web server. In fact, certain type of SSL and CA can be forged while everything appears to be legitimate. According to [5], secure browsing connection does virtually nothing to protect the users especially from the attackers that have knowledge on how the "secure" connections actually work. This paper develops an anti-web spoofing solution based on inspecting the URLs of fake web pages. This solution developed series of steps to check characteristics of websites Uniform Resources Locators (URLs). URLs of a phishing webpage typically have some unique characteristics that make it different from the URLs of a legitimate web page. Thus, URL is used in this paper to determine the location of the resource in computer networks.

C. Jian Maoi , Wenqian Tiani , Pei Li1 , Tao Wei , And Zhenkai Liang: "Phishing-Alarm: Robust and Efficient Phishing Detection via Page Component Similarity" –

Social networks have become one of the most popular platforms for users to interact with each other. Given the huge amount of sensitive data available in social network platforms, user privacy protection on social networks has become one of the most urgent research issues. As a traditional information stealing technique, phishing attacks still work in their way to cause a lot of privacy violation incidents. In a Web-based phishing attack, an attacker sets up scam Web pages (pretending to be an important Website such as a social network portal) to lure users to input their private information, such as passwords, social security numbers, credit card numbers, and so on. In fact, the appearance of Web pages is among the most important factors in deceiving users, and thus, the similarity among Web pages is a critical metric for detecting phishing Websites. In this paper, we present a new

solution, called Phishing-Alarm, to detect phishing attacks using features that are hard to evade by attackers. We present an algorithm to quantify the suspiciousness ratings of Web pages based on the similarity of visual appearance between the Web pages. Since cascading style sheet (CSS) is the technique to specify page layout across browser implementations, our approach uses CSS as the basis to accurately quantify the visual similarity of each page element. As page elements do not have the same influence on pages, we base our rating method on weighted page-component similarity. We prototyped our approach in the Google Chrome browser. Our large-scale evaluation using real-world websites shows the effectiveness of our approach. The proof of concept implementation verifies the correctness and accuracy of our approach with a relatively low performance overhead.

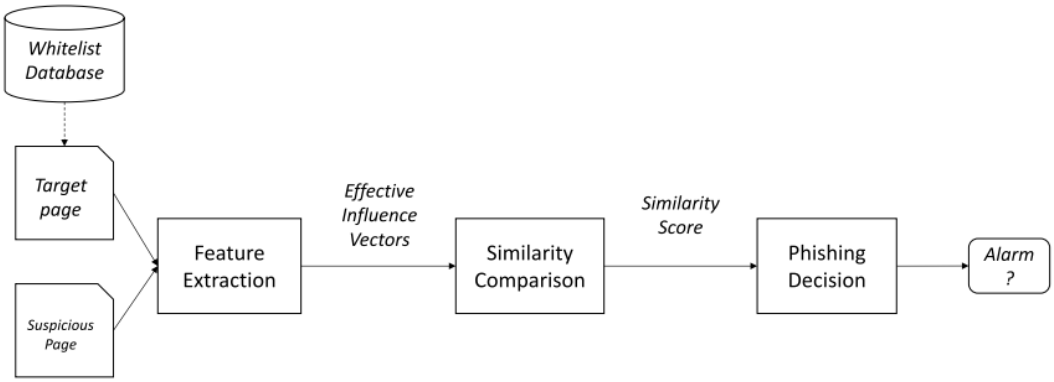


Fig. 2.2 Phishing detection via page component similarity

Our approach works in three phases: feature extraction, similarity computation, and phishing decision. In the first phase, feature extraction, given a suspicious page Ps, we extract its CSS structure CSS(Sus), and convert it into the infection vector to represent the static feature of page PS's visual layout. We also maintain a white-list database, which contains popular web pages targeted by phishing attacks. We extract the page features from the database using the same method. In the second phase, based on the influence vectors, we match the similarity between the suspicious page and the pages in the whitelist database. Finally, we make the decision by comparing the pages' similarity scores to a pre-set threshold. If the similarity scores are beyond and there exist other clues indicating that two testing pages are different, the suspicious page will be considered as a phishing page another subset such as degree of positive of negative feelings, current concerns such as degree of leisure, Spoken features such as degree of assent, and punctuation such as number

of colons. In addition to the raw features collected from LIWC, we also incorporated additional features (230 features) based on various combinations of the raw features.

D. Mohammed Nazim Feroz, "Phishing URL detection using URL Ranking" -

The openness of the Web exposes opportunities for criminals to upload malicious content. In fact, despite extensive research, email-based spam filtering techniques are unable to protect other web services. Therefore, a counter measure must be taken that generalizes across web services to protect the user from phishing host URLs. This paper describes an approach that classifies URLs automatically based on their lexical and host-based features. Clustering is performed on the entire dataset and a cluster ID (or label) is derived for each URL, which in turn is used as a predictive feature by the classification system. Online URL reputation services are used in order to categorize URLs and the categories returned are used as a supplemental source of information that would enable the system to rank URLs. The classifier achieves 93-98% accuracy by detecting many phishing hosts, while maintaining a modest false positive rate. URL clustering, URL classification, and URL categorization mechanisms work in conjunction to give URLs a rank.

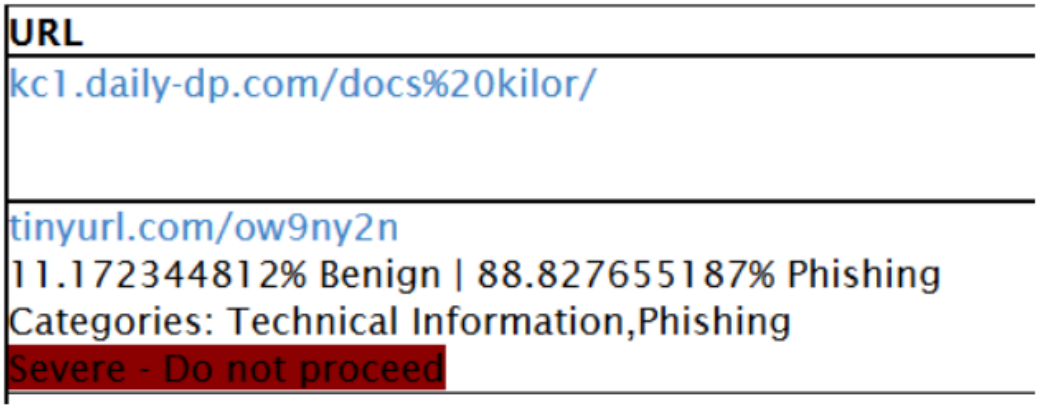


Fig. 2.3 Phishing detection using URL ranking

For the example given in the Fig. 2.3, URL categorization retrieved two categories 'Technical Information', and 'Phishing'; yielding Internal Scale = Red. The URL receives the rank 'Severe' based on Rule1 from the Rules at Internal Scale=Red. For the URL above, AS numbers of the website, mail server, and name server are different. IP prefixes of the

website, mail server, and name server are different. These among other factors contributed to the URL being classified as phishing.

E. Muhammet Baykara: "Detection of phishing attacks"-

Phishing is a form of cybercrime where an attacker imitates a real person / institution by promoting them as an official person or entity through e-mail or other communication mediums. In this type of cyber-attack, the attacker sends malicious links or attachments through phishing emails that can perform various functions, including capturing the login credentials or account information of the victim. These emails harm victims because of money loss and identity theft. In this study, a software called "Anti Phishing Simulator" was developed, giving information about the detection problem of phishing and how to detect phishing emails. With this software, phishing and spam mails are detected by examining mail contents. Classification of spam words added to the database by Bayesian algorithm is provided.

A simple flowchart of the implemented application is given in Fig. 2.4. Today, an e-mail can be found in primitive ways whether it is a phishing message or not. For this are looked where this email came from, whether a link with the message matches the actual website, whether the email or referrer web site is using some emotional or exciting words to get a response, whether it is spelling or grammar errors in the email or on the website. However, many people pay attention to this point unconsciously entering the links given to others' accounts.

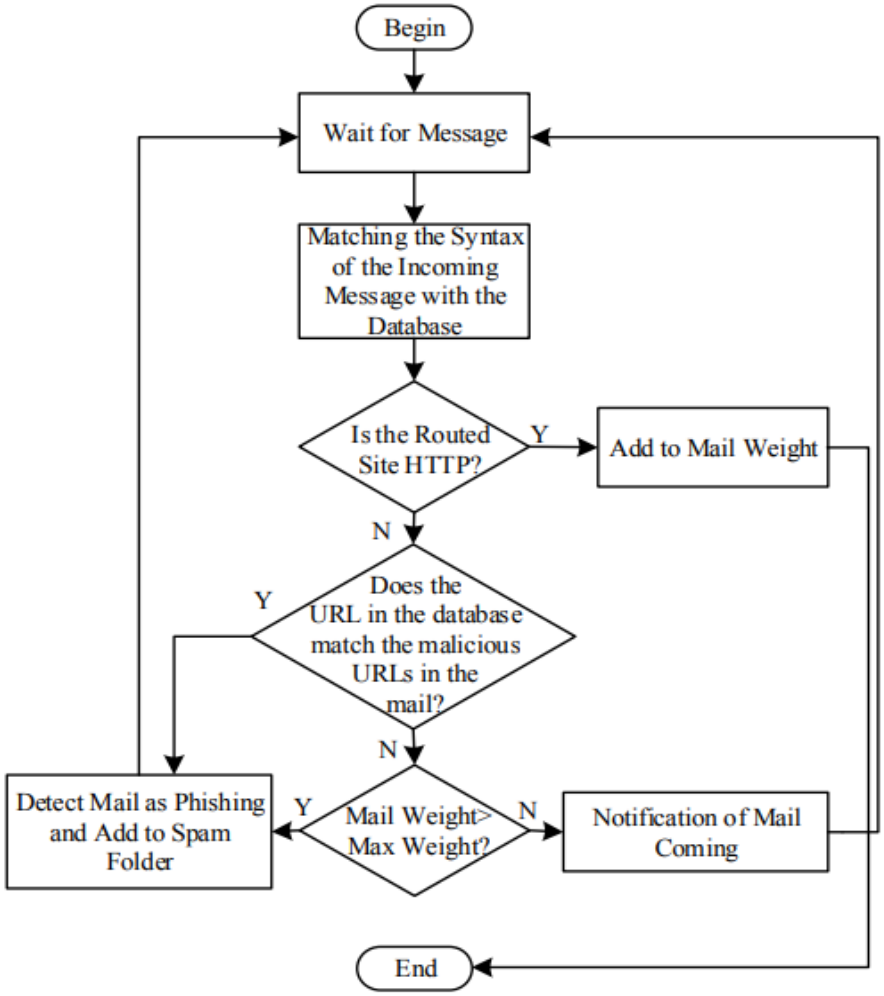


Fig. 2.4 Flowchart of Anti-Phishing Simulator

In this work, "Anti Phishing Simulator" decides whether a message is phishing thanks to the Bayesian classification algorithm and the scores added to the database. It is instantly perceived as a spam message by the words that are exciting, phrases that increase the desire for shopping, and which contain unwanted content. In addition, these spam mails can be viewed from the spam section. At the same time, it is possible to add an unwanted site URL address or an unwanted word to the database with the "add spam" feature. The page's html code is displayed with the "URL control" feature for those who have mastered the computer programming language. In this way, it can be checked whether the links in the page are valid or not. Although the database is very large, there is "add spam" feature to manage it on demand. In this way, the user can filter out messages that are not really spam, but that he does not want to see.

Chapter 3

THEORETICAL BACKGROUND

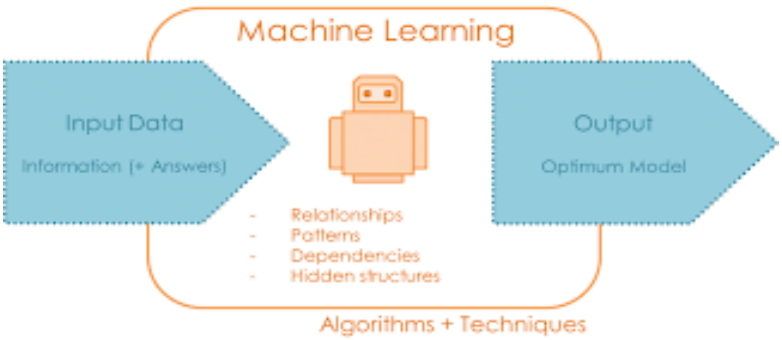
The theoretical concepts and algorithms used in the development of the proposed system are explained in the subsequent sections.

3.1 Machine Learning

Machine learning is a field of technology that uses applied mathematics techniques to supply the systems the power to be told with information, with no specific programming. It has evolved from the study of pattern recognition and process learning theory in AI and it explores the analysis of algorithms which will learn from and build predictions on information.

Machine learning is closely associated with process statistics that additionally focuses on prediction-making. Machine learning may also be unsupervised that is employed to find out and establish activity profiles for varied entities and find purposeful anomalies. Within the scope of information analytics, machine learning is used to plot complicated models and algorithms that facilitate in prediction and it is also known as predictive analytics.

Fig. 3.1 Machine Learning Process



3.1.1 Machine Learning Tasks

Machine learning tasks are classified into two broad categories, depending on whether there is feedback available to the learning system:

• **Supervised learning**: The computer is presented with sample inputs and their desired outputs, and the goal is to learn a general rule that maps inputs to outputs. As special cases, the input signal can be only partially available, or restricted to special feedback:

- Semi-supervised learning: The computer is given an incomplete training set with some (or many) of the target outputs missing.
- Active learning: The computer can only obtain training labels for a limited set, and also has to optimize its choice of objects to acquire labels.
- Reinforcement learning: The training data is given only as feedback to the program's actions in a dynamic environment, such as driving a vehicle or playing a game against an opponent.

• **Unsupervised learning**: No labels are given to the learning algorithm, as it is left on its own to find structure in the input. Unsupervised learning can be a goal in itself (discovering hidden patterns in data).

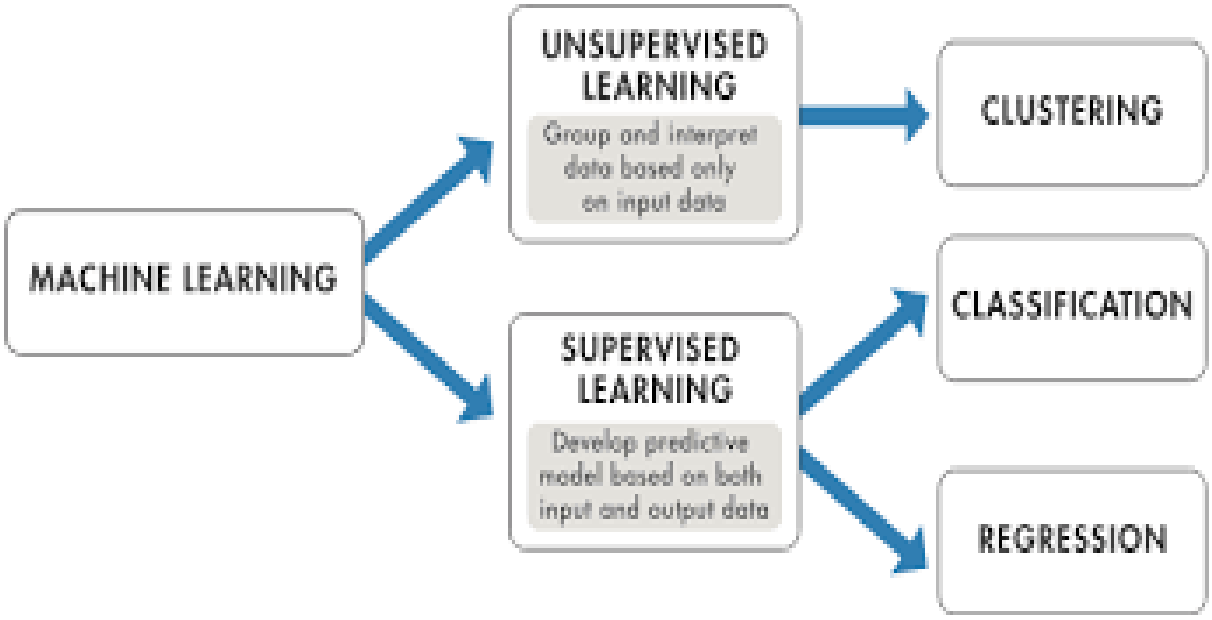


Fig. 3.2 Categories of Machine Learning tasks

3.1.2 Approaches to Machine Learning

- **Decision tree learning**

  Decision tree learning uses decision tree as a predictive model, which maps observations about an item to conclude about the target value.

- **Association rule learning**

  Association rule learning is used for discovering interesting relations between variables in large databases.

- **Artificial neural networks**

  An artificial neural network (ANN) learning algorithm, usually called "neural network" (NN), is a learning algorithm where computations are structured in terms of an interconnected group of artificial neurons and processes information using a connectionist approach to computation.

- **Deep learning**

  Deep learning consists of multiple hidden layers in an artificial neural network and it tries to model the way the human brain processes light and sound into vision and hearing.

- **Support vector machines**

  Support vector machines (SVMs) are a set of related supervised learning methods for classification and regression. A set of training examples will be given, where each belongs to one of two categories and the training algorithm builds a model that predicts whether a new example falls into one category or the other.

- **Clustering**

  Cluster analysis is grouping of a set of observations into clusters so that observations within the same cluster are similar according to some criteria, while observations from different clusters are dissimilar. Clustering is a unsupervised learning approach, and a used for statistical data analysis.

- **Bayesian networks**

  A Bayesian network, or directed acyclic graphical model is a probabilistic graphical model that represents a set of random variables and their conditional independencies via a directed acyclic graph (DAG). Efficient algorithms are used that perform inference and learning.

- **Representation learning**

  Several unsupervised learning algorithms aim at discovering better representations of the inputs provided during training. Representation learning algorithms are used to preserve the information in the input and transform it in a way that makes it useful, often as a pre-processing step before classification or predictions.

- **Sparse dictionary learning**

  Data is represented as a linear combination of basic functions, and the coefficients are assumed to be sparse. Learning a dictionary with sparse representations is NP-hard and also difficult to solve approximately. A popular heuristic method for sparse dictionary learning is K-SVD.

  Sparse dictionary learning is used in several contexts. In classification, the problem is to determine which classes a new data belongs to. It can be applied in image de-noising where a clean image patch can be sparsely represented by an image dictionary.

- **Rule-based machine learning**

  Rule-based machine learning is a general method for any machine learning process that identifies, learns, or evolves rules to store, manipulate or apply knowledge. The defining characteristic is the identification and utilization of a set of relational rules that collectively represent the knowledge captured by the system.

- **Learning classifier systems**

  Learning classifier systems (LCS) are a family of rule-based machine learning algorithms that combine a discovery component with a learning component. The systems try to identify a set of context-dependent rules that collectively store and apply knowledge for predictions.

3.2 Applications of Machine Learning

- **Image Recognition**

  One of the most common uses of machine learning is image recognition. There are many situations where the object can be classified as a digital image. For digital images, the measurements describe the outputs of each pixel in the image.

In a black and white image, the intensity of each pixel is one measurement. So, if a black and white image has N*N pixels, the total number of pixels and the measurement is N2. In a coloured image, each pixel considered to provide 3 measurements to the intensities of 3 main colour components, RGB. So, in N*N coloured image there are 3 N2 measurements.

• For face detection – The categories are face versus no face present. There can be a separate category for each person in a database of several individuals.

• For character recognition – A piece of writing can be segmented into smaller images, each containing a single character. The categories may contain the 26 letters of the English alphabet, the 10 digits, and some special characters.

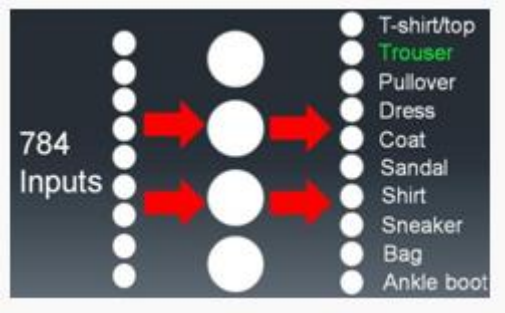


Fig. 3.3 Image recognition for fashion using Machine Learning

- **Speech Recognition**

  Speech recognition (SR) is the translation of spoken words into text. In speech recognition, a software application recognizes spoken words. The measurement in this application is a set of numbers that represent the speech signal. The signal can be segmented into portions that contain distinct words. In each segment, the speech signal can be represented by the intensities or energy in different time-frequency bands. Speech recognition applications include voice user interfaces. It also uses as simple data entry, preparation of structured documents, speech-to-text processing, and plane.

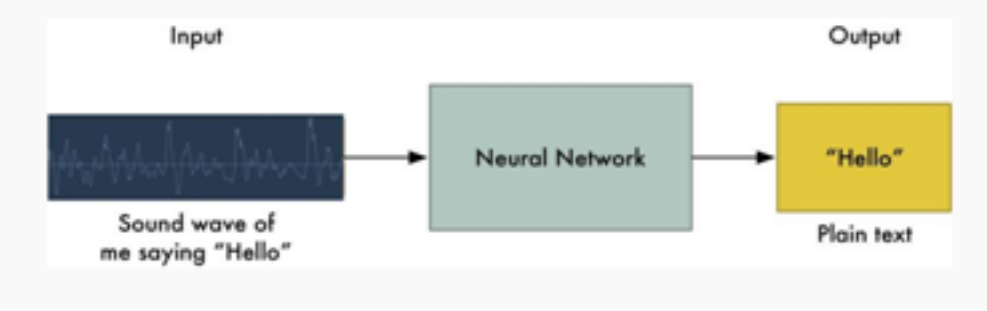


Fig. 3.4 Speech recognition using Neural Network

- **Medical Diagnosis**

  Machine Learning provides methods that can help solving diagnostic and prognostic problems in medical domains. It is used for the analysis of the importance of clinical parameters and of their combinations for prognosis in order to extract medical knowledge. Machine Learning is also used for data analysis, such as detection of regularities in the data by dealing with imperfect data, interpretation of continuous data used in the ICU, and for intelligent alarming resulting in effective monitoring.

  In medical diagnosis, the main interest is to establish the existence of a disease followed by its accurate identification. There is a separate category for each disease under consideration and one category for cases where no disease is present

  and machine learning improves the accuracy of diagnosis by analysing data of patients.

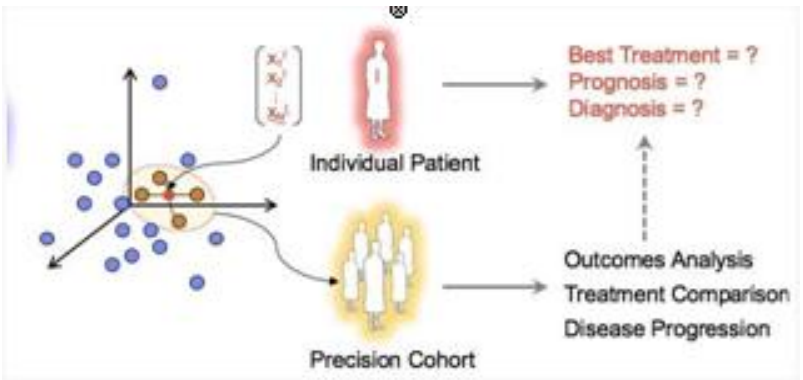


Fig. 3.5 Machine Learning for treatment of diseases

- **Statistical Arbitrage**

  In finance, statistical arbitrage refers to the automated trading strategies that are typically of a short term and involve many securities. The user tries to implement a trading algorithm for a set of securities based on quantities such as historical correlations and general economic variables. These measurements can be applied as a classification or estimation problem.

  The machine learning methods can be applied to obtain an index arbitrage strategy. Linear regression and support vector regression (SVR) are employed onto the prices of an exchange-traded fund and a stream of stocks. By using principal component analysis (PCA) in dimension reduction of feature space, the benefits and the issues are observed in the application of SVR.

Fig. 3.6 Statistical Arbitrage trading pairs

- **Learning Associations**

  Learning association is the process of developing insights into various associations between products. It may explain how seemingly unrelated products may reveal an association to one another.

  One application of machine learning is the association between the products people buy, which is also known as basket analysis. If a buyer buys „X", would he or she force to buy „Y" because of a relationship that can identify between them. These relationships help in suggesting the associated product to the customer.

Fig. 3.7 Example of Association Rules

- **Classification**

  Classification is the process of grouping everyone from the population under study in many classes. It helps analysts to use measurements of an object to identify the category to which that object belongs. To establish an efficient rule, analysts use the data which consists of many examples of objects with their correct classification. For example, consider a problem of spam detection in email. In such cases, an email can be either a spam or not a spam, so there are two classes in this problem and an email is classified into spam or non-spam classes.

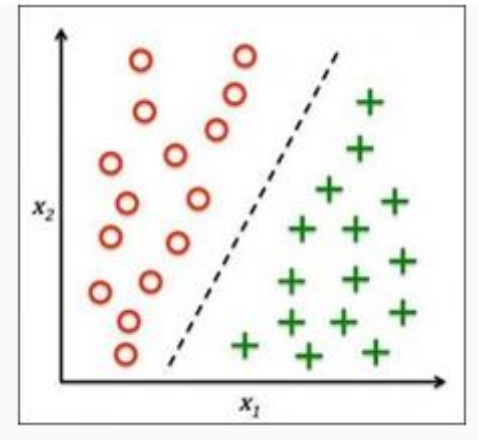


Fig. 3.8 Linear Classification

- **Prediction**

  Prediction is one of the most important machine learning algorithms which helps the business to take required decision on time.

Consider the example of a bank computing the probability of any of loan applicants faulting the loan repayment. To compute the probability of the fault, the system first classifies the available data in certain groups which is described by a set of rules prescribed by the analysts. Then the probability can be computed across all sectors for varied purposes.

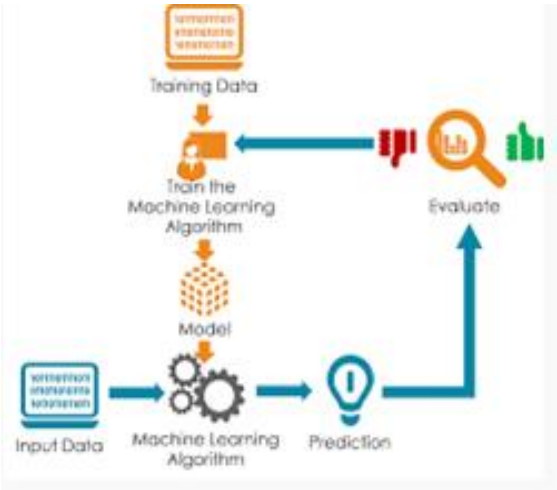


Fig. 3.9 Prediction process

- **Extraction**

Information Extraction (IE) is an application of machine learning which deals with the extraction of structured information from unstructured data such as web pages, articles, blogs, business reports, and e-mails. The relational database maintains the output produced by the information extraction. The process of extraction takes input as a set of documents and produces a structured data. This output is in summarized form such as excel sheet and table in a relational database.

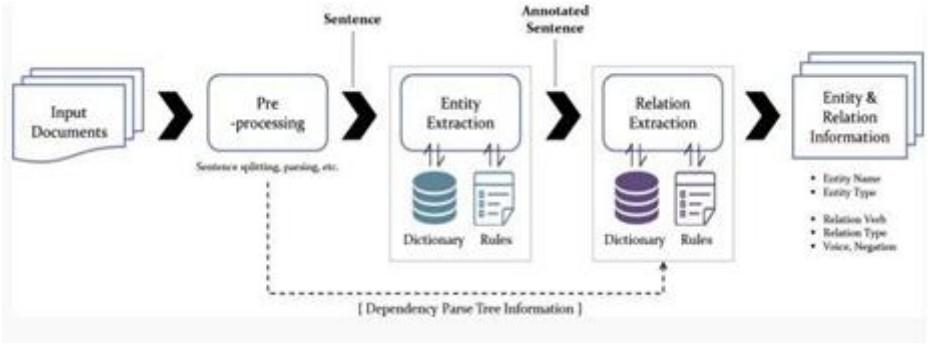


Fig. 3.10 Process of extraction

- **Regression**

  In regression, the principle of machine learning is used to optimize the parameters and to reduce the approximation error and calculate the closest possible outcome.

  Assume that x= x_1, x_2, x_3, … x_n are the input variables and y is the outcome variable. In this case, machine learning technology produces the output (y) based on the input variables (x). A model can be used to express the relationship between various parameters as below: Y=g(x) where g is a function that depends on specific characteristics of the model.

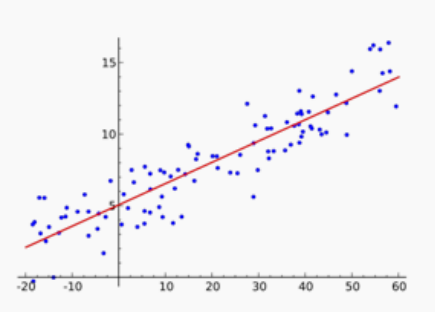


Fig. 3.11 Linear Regression

3.3 Features of the websites

3.3.1 Address Bar Based Features

i] Using the IP Address

If an IP address is used as an alternative of the domain name in the URL, such as "http://125.98.3.123/fake.html", users can be sure that someone is trying to steal their personal information. Sometimes, the IP address is even transformed into hexadecimal code as shown in the following link "http://0x58.0xCC.0xCA.0x62/2/paypal.ca/index.html".

Rule: IF{$If\ The\ Domain\ Part\ has\ an\ IP\ Address \rightarrow Phishing$

$Otherwise \rightarrow Legitimate$

ii] Long URL to Hide the Suspicious Part

Phishers can use long URL to hide the doubtful part in the address bar. For example:

http://federmacedoadv.com.br/3f/aze/ab51e2e319e51502f416dbe46b773a5e/?cmd=_home&dispatch=11004d58f5b74f8dc1e7c2e8dd4105e811004d58f5b74f8dc1e7c2e8dd4105e8@phishing.website.html

To ensure accuracy of our study, we calculated the length of URLs in the dataset and produced an average URL length. The results showed that if the length of the URL is greater than or equal 54 characters then the URL classified as phishing. By reviewing our dataset, we were able to find 1220 URLs lengths equals to 54 or more which constitute 48.8% of the total dataset size.

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

We have been able to update this feature rule by using a method based on frequency and thus improving upon its accuracy.

iii] Using URL Shortening Services "TinyURL"

URL shortening is a method on the "World Wide Web" in which a URL may be made considerably smaller in length and still lead to the required webpage. This is accomplished by means of an "HTTP Redirect" on a domain name that is short, which links to the webpage that has a long URL. For example, the URL "http://portal.hud.ac.uk/" can be shortened to "bit.ly/19DXSk4".

Rule: IF$\{TinyURL \rightarrow Phishing\ Otherwise \rightarrow Legitimate$

iv] URL's having "@" Symbol

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

Rule: IF $\{Url\ Having\ @\ Symbol \rightarrow Phishing\ Otherwise \rightarrow Legitimate$

v] Redirecting using "//"

The existence of "//" within the URL path means that the user will be redirected to another website. An example of such URL's is: "http://www.legitimate.com//http://www.phishing.com". We examine the location where the "//" appears. We find that if the URL starts with "HTTP", that means the "//" should

appear in the sixth position. However, if the URL employs "HTTPS" then the "//" should appear in seventh position.

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

vi] Adding Prefix or Suffix Separated by (-) to the Domain

The dash symbol is rarely used in legitimate URLs. Phishers tend to add prefixes or suffixes separated by (-) to the domain name so that users feel that they are dealing with a legitimate webpage. For example, http://www.Confirme-paypal.com/.

Rule: IF $\{Domain\ Name\ Part\ Includes\ (-)\ Symbol \rightarrow Phishing$

$Otherwise \rightarrow Legitimate$

vii] Sub Domain and Multi Sub Domains

Let us assume we have the following link: http://www.hud.ac.uk/students/. A domain name might include the country-code top-level domains (ccTLD), which in our example is "uk". The "ac" part is shorthand for "academic", the combined "ac.uk" is called a second-level domain (SLD) and "hud" is the actual name of the domain. To produce a rule for extracting this feature, we firstly must omit the (www.) from the URL which is in fact a sub domain. Then, we must remove the (ccTLD) if it exists. Finally, we count the remaining dots. If the number of dots is greater than one, then the URL is classified as "Suspicious" since it has one sub domain. However, if the dots are greater than two, it is classified as "Phishing" since it will have multiple sub domains. Otherwise, if the URL has no sub domains, we will assign "Legitimate" to the feature.

Rule: IF $\{Dots\ In\ Domain\ Part = 1 \rightarrow Legitimate\ Dots\ In\ Domain\ Part = 2 \rightarrow Suspicious\ \ Otherwise \rightarrow Phishing$

viii] HTTPS (Hyper Text Transfer Protocol with Secure Sockets Layer)

The existence of HTTPS is very important in giving the impression of website legitimacy, but this is clearly not enough. The authors in (Mohammad, Thabtah and McCluskey 2012) (Mohammad, Thabtah and McCluskey 2013) suggest checking the certificate assigned with HTTPS including the extent of the trust certificate issuer, and the certificate age. Certificate Authorities that are consistently listed among the top trustworthy names include: "GeoTrust, GoDaddy, Network Solutions, Thawte, Comodo, Doster and VeriSign".

Furthermore, by testing out our datasets, we find that the minimum age of a reputable certificate is two years.

Rule: IF{$Use\ https\ and\ Issuer\ Is\ Trusted\ and\ Age\ of\ Certificate \geq 1\ Years \rightarrow Legitimate\ \ Using\ https\ and\ Issuer\ Is\ Not\ Trusted \rightarrow Suspicious \quad Otherwise \rightarrow Phishing$

ix] Domain Registration Length

Since 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.

Rule: IF{$Domains\ Expires\ on \leq 1\ years \rightarrow Phishing \quad Otherwise \rightarrow Legitimate$

x] 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.

Rule: IF{$Favicon\ Loaded\ From\ External\ Domain \rightarrow Phishing \quad Otherwise \rightarrow Legitimate$

xi] Using Non-Standard Port

This feature is useful in validating if a service (e.g. HTTP) is up or down on a specific server. In the aim of controlling intrusions, it is much better to merely open ports that you need. Several firewalls, Proxy and Network Address Translation (NAT) servers will, by default, block all or most of the ports and only open the ones selected. If all ports are open, phishers can run almost any service they want and as a result, user information is threatened. The most important ports and their preferred status are shown in Table 2.

Rule: IF{$Port\ \#\ is\ of\ the\ Prefered\ Status \rightarrow Phishing\ Otherwise \rightarrow Legitimate$

xii] The Existence of "HTTPS" Token in the Domain Part of the URL

The phishers may add the "HTTPS" token to the domain part of a URL in order to trick users.

Example, http://https-www-paypal-it-webapps-mpp-home.soft-hair.com/.

Rule: IF{$Using\ HTTP\ Token\ in\ Domain\ Part\ of\ The\ URL \rightarrow\ Phishing$

$Otherwise \rightarrow\ Legitimate$

3.3.2 Abnormal Based Features

i] Request URL

Request URL examines whether the external objects contained within a web page such as images, videos and sounds are loaded from another domain. In legitimate webpages, the webpage address and most of objects embedded within the webpage are sharing the same domain.

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

ii] URL of Anchor

An anchor is an element defined by the <a> tag. This feature is treated exactly as "Request URL". However, for this feature we examine:

1. If the <a> tags and the website have different domain names. This is similar to request URL feature.
2. If the anchor does not link to any webpage, e.g.:
   A. <a href="#">
   B. <a href="#content">
   C. <a href="#skip">
   D. <a href="JavaScript ::void(0)">

Rule: IF{$\%\ of\ URL\ Of\ Anchor\ < 31\% \rightarrow\ Legitimate\ \%\ of\ URL\ Of\ Anchor \geq 31\%\ And \leq 67\% \rightarrow\ Suspicious\ Otherwise \rightarrow\ Phishing$

iii] Links in <Meta>, <Script> and <Link> tags

Given that our investigation covers all angles likely to be used in the webpage source code, we find that it is common for legitimate websites to use <Meta> tags to offer metadata

about the HTML document; <Script> tags to create a client side script; and <Link> tags to retrieve other web resources. It is expected that these tags are linked to the same domain of the webpage.

Rule: IF{$\%\ of\ Links\ in\ "<Meta>","<Script>"\ and\ "<Link>"<17\% \rightarrow Legitimate\ \%\ of\ Links\ in\ <Meta>","<Script>"\ and\ "<Link>"\geq 17\%\ And \leq 81\% \rightarrow Suspicious\ Otherwise \rightarrow Phishing$

iv] Server Form Handler (SFH)

SFHs that contain an empty string or "about: blank" are considered doubtful because an action should be taken upon the submitted information. In addition, if the domain name in SFHs is different from the domain name of the webpage, this reveals that the webpage is suspicious because the submitted information is rarely handled by external domains.

Rule: IF{$SFH\ is\ "about:blank"\ Or\ Is\ Empty \rightarrow Phishing$ $SFH\ Refers\ To\ A\ Different\ Domain \rightarrow Suspicious$ $Otherwise \rightarrow Legitimate$

v] Submitting Information to Email

Web form allows a user to submit his personal information that is directed to a server for processing. A phisher might redirect the user's information to his personal email. To that end, a server-side script language might be used such as "mail()" function in PHP. One more client-side function that might be used for this purpose is the "mailto:" function.

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

vi] Abnormal URL

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

Rule: IF {$The\ Host\ Name\ Is\ Not\ Included\ In\ URL \rightarrow Phishing$ $Otherwise \rightarrow Legitimate$

3.3.3 HTML and JavaScript Based Features

i] Website Forwarding

The fine line that distinguishes phishing websites from legitimate ones is how many times a website has been redirected. In our dataset, we find that legitimate websites have been redirected one-time max. On the other hand, phishing websites containing this feature have been redirected at least 4 times.

Rule: IF $\{\#ofRedirectPage \leq 1 \rightarrow Legitimate \; \#of\; Redirect\; Page \geq 2\; And < 4 \rightarrow Suspicious \quad Otherwise \rightarrow Phishing$

ii] Status Bar Customization

Phishers may use JavaScript to show a fake URL in the status bar to users. To extract this feature, we must dig-out the webpage source code, particularly the "onMouseOver" event, and check if it makes any changes on the status bar.

Rule: IF$\{onMouseOver\; Changes\; Status\; Bar \rightarrow Phishing \quad It\; Doesn't\; Change\; Status\; Bar \rightarrow Legitimate$

iii] Disabling Right Click

Phishers use JavaScript to disable the right-click function, so that users cannot view and save the webpage source code. This feature is treated exactly as "Using onMouseOver to hide the Link". Nonetheless, for this feature, we will search for event "event.button==2" in the webpage source code and check if the right click is disabled.

Rule: IF$\{Right\; Click\; Disabled \rightarrow Phishing \quad Otherwise \rightarrow Legitimate$

iv] Using Pop-up Window

It is unusual to find a legitimate website asking users to submit their personal information through a pop-up window. On the other hand, this feature has been used in some legitimate websites and its main goal is to warn users about fraudulent activities or broadcast a welcome announcement, though no personal information was asked to be filled in through these pop-up windows.

Rule: IF $\{Popup\; Window\; Contains\; Text\; Fields \rightarrow Phishing$

$Otherwise \rightarrow Legitimate$

v] IFrame Redirection

IFrame is an HTML tag used to display an additional webpage into one that is currently shown. Phishers can make use of the "iframe" tag and make it invisible i.e. without frame borders. In this regard, phishers make use of the "frameBorder" attribute which causes the browser to render a visual delineation.

Rule: IF $\{Using\ iframe \rightarrow Phishing \qquad Otherwise \rightarrow Legitimate$

3.3.4 Domain Based Features

i] Age of Domain

This feature can be extracted from WHOIS database (Whois 2005). Most phishing websites live for a short period of time. By reviewing our dataset, we find that the minimum age of the legitimate domain is 6 months.

Rule: IF $\{Age\ Of\ Domain \geq 6\ months \rightarrow Legitimate \qquad Otherwise \rightarrow Phishing$

ii] DNS Record

For phishing websites, either the claimed identity is not recognized by the WHOIS database (Whois 2005) or no records found for the hostname (Pan and Ding 2006). If the DNS record is empty or not found then the website is classified as "Phishing", otherwise it is classified as "Legitimate".

Rule: IF$\{no\ DNS\ Record\ For\ The\ Domain \rightarrow Phishing \quad Otherwise \rightarrow Legitimate$

iii] Website Traffic

This feature measures the popularity of the website by determining the number of visitors and the number of pages they visit. However, since phishing websites live for a short period of time, they may not be recognized by the Alexa database (Alexa the Web Information Company., 1996). By reviewing our dataset, we find that in worst scenarios, legitimate websites ranked among the top 100,000. Furthermore, if the domain has no traffic or is not recognized by the Alexa database, it is classified as "Phishing". Otherwise, it is classified as "Suspicious".

Rule: IF$\{Website\ Rank < 100{,}000 \rightarrow Legitimate \qquad Website\ Rank > 100{,}000 \rightarrow Suspicious \qquad Otherwise \rightarrow Phishing$

iv] PageRank

PageRank is a value ranging from "0" to "1". PageRank aims to measure how important a webpage is on the Internet. The greater the PageRank value the more important the webpage. In our datasets, we find that about 95% of phishing webpages have no PageRank. Moreover, we find that the remaining 5% of phishing webpages may reach a PageRank value up to "0.2".

Rule: IF{$PageRank < 0.2 \rightarrow Phishing$ $Otherwise \rightarrow Legitimate$

v] Google Index

This feature examines whether a website is in Google's index or not. When a site is indexed by Google, it is displayed on search results (Webmaster resources, 2014). Usually, phishing webpages are merely accessible for a short period and as a result, many phishing webpages may not be found on the Google index.

Rule: IF{$Webpage\ Indexed\ by\ Google \rightarrow Legitimate$ $Otherwise \rightarrow Phishing$

vi] Number of Links Pointing to Page

The number of links pointing to the webpage indicates its legitimacy level, even if some links are of the same domain (Dean, 2014). In our datasets and due to its short life span, we find that 98% of phishing dataset items have no links pointing to them. On the other hand, legitimate websites have at least 2 external links pointing to them.

Rule: IF{$\#Of\ Link\ Pointing\ to\ The\ Webpage = 0 \rightarrow Phishing$ $\#Of\ Link\ Pointing\ to\ The\ Webpage > 0\ and \leq 2 \rightarrow Suspicious$ $Otherwise \rightarrow Legitimate$

vii] Statistical-Reports Based Feature

Several parties such as Phish Tank (Phish Tank Stats, 2010-2012), and StopBadware (StopBadware, 2010-2012) formulate numerous statistical reports on phishing websites at every given period of time; some are monthly, and others are quarterly. In our research, we used 2 forms of the top ten statistics from Phish Tank: "Top 10 Domains" and "Top 10 IPs" according to statistical reports published in the last three years, starting in January 2010 to November 2012. Whereas for "StopBadware", we used "Top 50" IP addresses.

Rule: IF{$Host\ Belongs\ to\ Top\ Phishing\ IPs\ or\ Top\ Phishing\ Domains \rightarrow Phishing$ $Otherwise \rightarrow Legitimate$

Chapter 4

SYSTEM REQUIREMENT SPECIFICATION

A software requirements specification (SRS) is a detailed description of the software system to be developed. It portrays the functional and non-functional requirements and may also include a set of use cases that describe user interactions that the software must provide.

Software Requirement Specification is a fundamental part of the document as it is the root foundation of the software development process. At the basic level, it is an organization's understanding of a customer or potential client's expectations, requirements and dependencies at a point of time, before any design or development work.

4.1 Functional Requirements

A function is defined as a set of inputs, behaviour and the outputs. A functional requirement can be defined as a specification which defines the function of the concerned system. This may include specific functionality such as technical details, calculations, data manipulation and processing etc. The functional requirement documents the operations and the activities that a system should be able to perform. The system design contains details about implementing the functional requirement. Below listed are the functional requirements in this system:

- Data collection - The data set used in this project is obtained from Kaggle website and the dataset includes 11,055 rows of data and 31 columns (30 features columns and 1 target column).
- Data Pre-processing - The purpose of pre-processing is to convert raw data into a form that fits machine learning. Structured and clean data allows a data scientist to get more precise results from an applied machine learning model. The technique includes data formatting, cleaning, and sampling.
- Dataset splitting -A dataset used for machine learning should be partitioned into three subsets — training, test, and validation sets.

- Model training -After a data scientist has pre-processed the collected data and split it into train and test can proceed with a model training. This process entails "feeding" the algorithm with training data. An algorithm will process data and output a model that is able to find a target value (attribute) in new data an answer you want to get with predictive analysis. The purpose of model training is to develop a model. In this project, 75% of the dataset is considered as training set and the rest 25% of the dataset is considered as testing set.
- Model evaluation and testing -The goal of this step is to develop the simplest model able to formulate a target value fast and well enough. A data scientist can achieve this goal through model tuning. That's the optimization of model parameters to achieve an algorithm's best performance.

- Implementation of the model- After obtaining the accuracy of all the trained models, the most accurate model is implemented using a google chrome extension that raises an alert when a phishing website is visited.

4.2 Non-functional requirements

While functional requirements specify the behaviour of the system and tell us what the system is supposed to do, the non-functional requirements are mostly concerned with how the system is supposed to be. The system architecture contains details about implementing these non-functional requirements.

The non-functional requirements specify the criteria that are used to evaluate the operation of a system, instead of a specific behaviour. These requirements are also often termed as "quality attributes", "constraints" and "non-behavioural requirements".

The following is a list of non-functional requirements. The specific details will need to be defined by internal stakeholders.

- Response Time
- Availability
- Stability
- Maintainability
- Usability

4.3 System Requirements

The system requirements include Hardware and Software requirement, which are provided below:

4.3.1 Hardware Requirements

The hardware requirements include the requirements specification of the physical computer resources for a system to work efficiently. The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system.

- Processor : Any Processor above 500 MHz
- RAM : 4 GB
- Hard Disk : 4 GB
- Input device : Standard Keyboard and Mouse.
- Output device : VGA and High-Resolution Monitor.

4.3.2 Software Requirements

Operating System : Windows 7 or higher

Programming libraries : Python 3.6 and related libraries, JavaScript and related

SOFTWARE DESCRIPTION:

PYTHON

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace. It provides constructs that enable clear programming on both small and large scales.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

Python interpreters are available for many operating systems. CPython, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all its variant implementations. CPython is managed by the non-profit Python Software Foundation.

JAVASCRIPT

JavaScript (often shortened to **JS**) is a lightweight, interpreted, object-oriented language with first-class functions, and is best known as the scripting language for Web pages, but it's used in many non-browser environments as well. It is a prototype-based, multi-paradigm scripting language that is dynamic, and supports object-oriented, imperative, and functional programming styles.

JavaScript runs on the client side of the web, which can be used to design / program how the web pages behave on the occurrence of an event. JavaScript is an easy to learn and powerful scripting language, widely used for controlling web page behaviour.

Contrary to popular misconception, **JavaScript is not "Interpreted Java"**. In a nutshell, JavaScript is a dynamic scripting language supporting prototype-based object construction. The basic syntax is intentionally similar to both Java and C++ to reduce the number of new concepts required to learn the language. Language constructs, such as if statements, for and while loops, and switch and try ... catch blocks function the same as in these languages (or nearly so).

JavaScript can function as both a procedural and an object-oriented language. Objects are created programmatically in JavaScript, by attaching methods and properties to otherwise empty objects **at run time**, as opposed to the syntactic class definitions common in compiled languages like C++ and Java. Once an object has been constructed it can be used as a blueprint (or prototype) for creating similar objects.

JavaScript's dynamic capabilities include runtime object construction, variable parameter lists, function variables, dynamic script creation (via eval), object introspection (via for ... in), and source code recovery (JavaScript programs can decompile function bodies back into their source text).

Chapter 5

SYSTEM ANALYSIS

System Analysis is the method of understanding an existing problem, defining requirements and evaluating the best possible solution. It enables thinking about the organization and its concerned problems as well as the technologies which can help in coming up with solutions.

System Analysis can be seen as a great problem-solving technique that breaks down the system into component modules and evaluates each module to determine how efficiently they work individually and how well they interact with each other, in order to accomplish a specific task. The feasibility study plays a very important role in system analysis as it provides the target for design and development.

5.1 Feasibility Study:

Feasibility study can be understood as the assessment of the practicality of the proposed system or application. The feasibility study is also the key to evaluate the potential of a project's success. There are two most important to measure the feasibility which includes the total cost involved and the value to be attained on completion. Usually, a feasibility study always precedes the technical development and implementation.

Risk analysis is very closely related to feasibility study. If there is a greater project risk, the feasibility of creating quality software is thus reduced. When we talk about feasibility study at depth, there are three primary areas of interest in this context:

1. Performance Analysis
2. Technical Analysis
3. Economic Analysis

All the three types of feasibility tests are useful in determining the overall practicality or feasibility of the proposed system. These tests enable to evaluate whether the considered solution to satisfy the requirements is practical and workable in the software developed. Thus, it is important to perform these tests individually.

5.1.1 Performance Analysis

The software system is made to run in suitable environment. The most important aim of performance analysis is to provide the end user with the required functionality of the system. The user must obtain the desirable results in an accurate and time-efficient manner. The process of performance definition should be carried out in parallel with performance analysis to set the milestone for efficient performance. The goal of development process to achieve requirements satisfaction is to carry out performance analysis as early as possible.The aim of the present system is to predict phishing websites in a more efficient and smarter way.

The benchmark of performance for our system can be set by the following performance definition:

- ✓ The user should obtain relevant results as per the situation demand.
- ✓ The user should get accurate prediction of the phishing website.
- ✓ The user should get an alert message as soon as they visit a phishing website.
- ✓ The output of the system should give the user a clear idea of whether the website is phishing website or not.
- ✓ The system should benefit the user such that it alerts the user about visiting any phishing website and not to be tricked by the attackers.

5.1.2 Technical Analysis

This feasibility test evaluates whether the proposed system will work and give results when it is fully developed and installed. It also analyses the presence of any obstacles in the implementation of the software application. After all a system can, be termed as efficient only if it can overcome all the technical requirements of the user.The technical modules and software required to implement this project are freely available. The project is

implemented using Python and other machine learning algorithms and Javascript. Since the resources are available, it can be concluded that the system is technically feasible.

5.1.3 Economic Analysis:

The economic analysis is performed to estimate the development cost weighed against the benefits which will be derived from the developed system.

For this project, only a computer system with Internet connection is required. The features desired from the system can be uncovered by any operating system. For running this system, a Mac/Ubuntu/Windows operating system installed on our machine is used. Thus, it can be concluded that the system is economically feasible.

Chapter 6

SYSTEM DESIGN

6.1 System Architecture

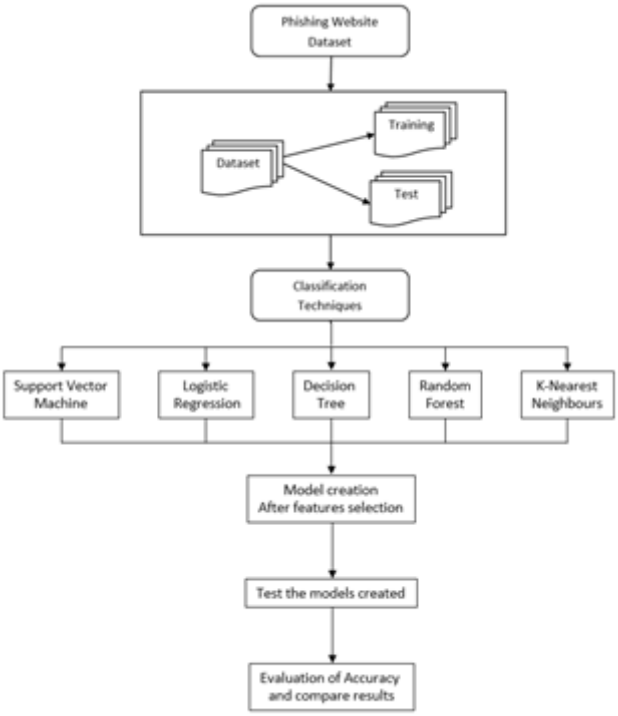


Fig. 6.1 Architecture of the detection system

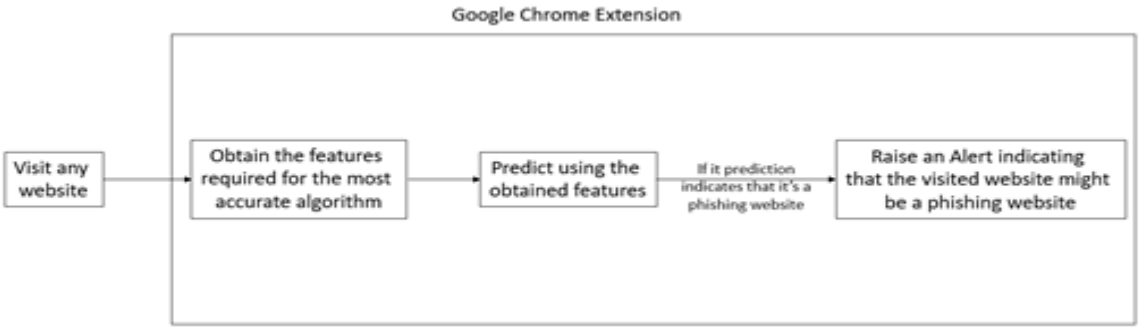


Fig. 6.2 Phishing alert using google chrome extension

Fig. 6.1 illustrates the proposed phishing detection process, which includes two phases: training and detection. In the training phase, a classifier is generated using URLs of phishing sites and legitimate sites collected in 1) secure, account, websrc, sign-in, confirm, login 2) index, includes, content, images, admin, file_doc, paypal, login advance. The collected URLs are transmitted to the feature extractor, which extracts feature values through the predefined URL-based features. The extracted features are stored as input and passed to the classifier generator, which generates a classifier by using the input features and the machine learning algorithm. In the detection phase, the classifier determines whether a requested site is a phishing site. When a page request occurs, the URL of the requested site is transmitted to the feature extractor, which extracts the feature values through the predefined URL-based features. Those feature values are inputted to the classifier. The classifier determines whether a new site is a phishing site based on learned information. It then alerts the page-requesting user about the classification result.

6.2 Phishing website detection based on machine learning classifiers with features selection

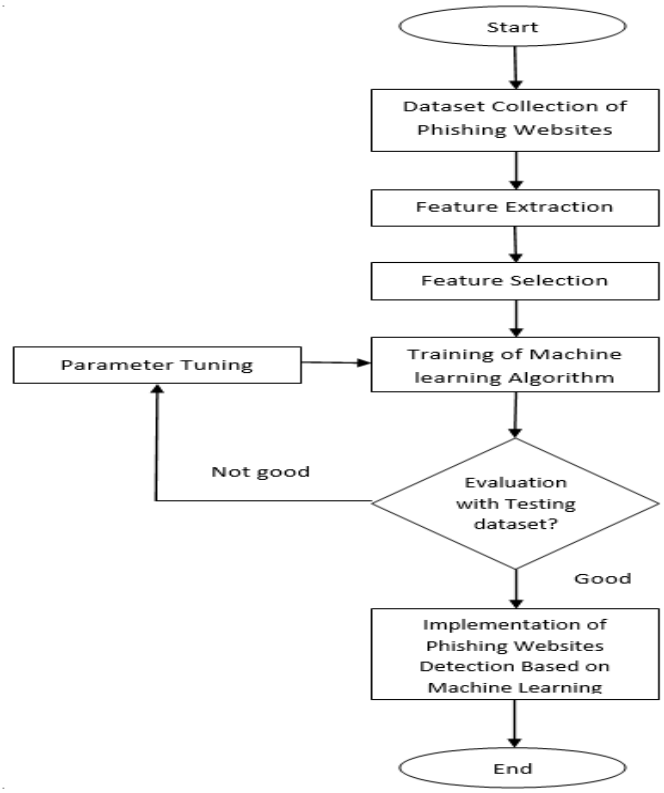


Fig. 6.3 Flowchart of phishing website detection based on features selection

6.3 Machine Learning Algorithms

To determine a classifier with the best performance for using URL-based features, we employed several machine learning algorithms: Support Vector Machine (SVM), K-Nearest Neighbours (KNN), Logical Regression, Decision Tree and Random Forest.

6.3.1 Support Vector Machine (SVM)

SVM is a classification method that was introduced in 1992 by Boser, Gyon, and Vapnik. It is a statistical learning algorithm that classifies samples using a subset of the training samples, called support vectors. SVM is built on the structural risk minimization principle for seeking a decision surface that can separate data points into two classes with a minimal margin between them. The advantage of SVM is its capability of learning in spare high-dimensional spaces with very few training samples.

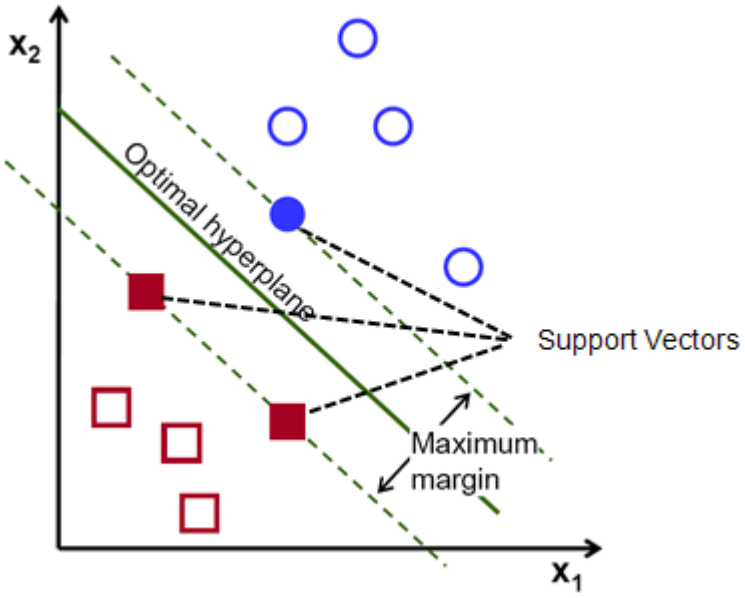


Fig. 6.4 Support Vector Machine Classification

6.3.2 Decision Tree (DT)

Decision tree is a classification method that was introduced in 1992 by Quinlan. It creates a tree form for classifying samples. Each internal node of the tree corresponds to a feature,

and the edges from the node separate the data based on the value of the feature. Decision tree includes a decision area and leaf node. The decision area checks the condition of the samples and separates them into each leaf node or the next decision area. The decision tree is very fast and easy to implement; however, it has the risk of overfitting.

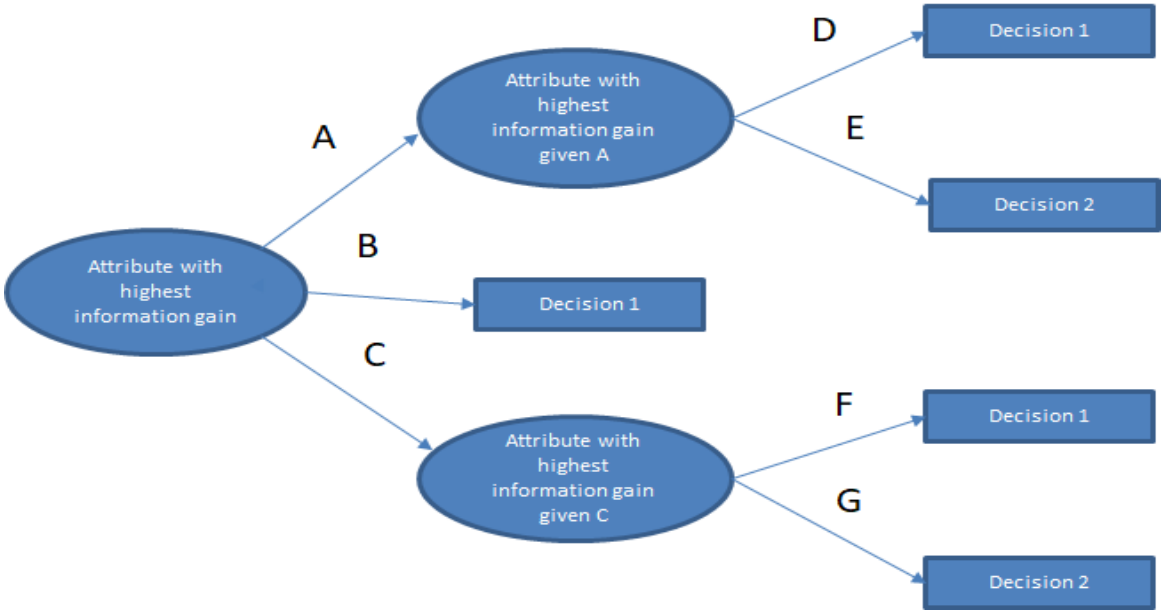


Fig. 6.4 Flow diagram of Decision Tree

6.3.3 K-Nearest Neighbours (KNN)

KNN is a non-parametric classification algorithm. It has been successfully applied to various information-retrieval problems. It classifies the input data using k training data that is similar to the input data. KNN uses Euclidean distance to calculate the similarity between the input and training samples. Its performance is determined by the choice of k; nevertheless, choosing a suitable k value is not easy.

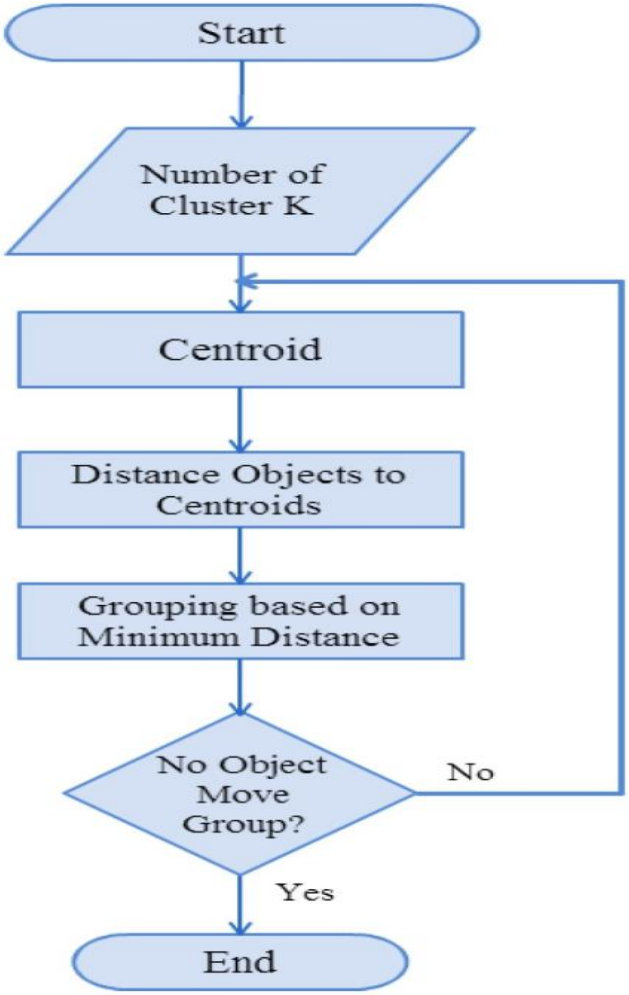


Fig. 6.5 Flowchart of KNN

6.3.4 Random Forest (RF)

Random forest is a classification method that combines many tree predictors; each tree depends on the values of a random vector that is independently sampled [19]. All trees in the forest have the same distribution. This algorithm can handle a large number of variables in the dataset; however, it lacks reproducibility because the process of forest building is random.

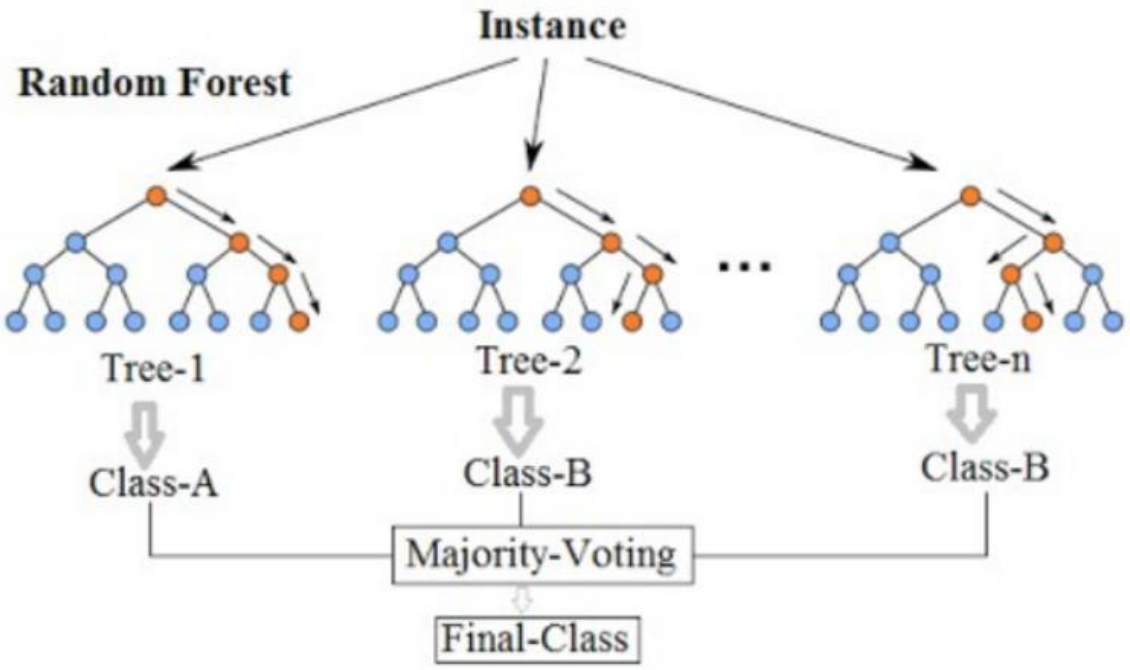


Fig. 6.6 Flow diagram of Random Forest

6.3.5 Logistic Regression (LR)

Logistic regression is named for the function used at the core of the method, the logistic function. The logistic function, also called the sigmoid function was developed by statisticians to describe properties of population growth in ecology, rising quickly and maxing out at the carrying capacity of the environment. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

1 / (1 + e^-value), where e is the base of the natural logarithms (Euler's number or the EXP() function in your spreadsheet) and value is the actual numerical value that you want to transform. Below is a plot of the numbers between -5 and 5 transformed into the range 0 and 1 using the logistic function.

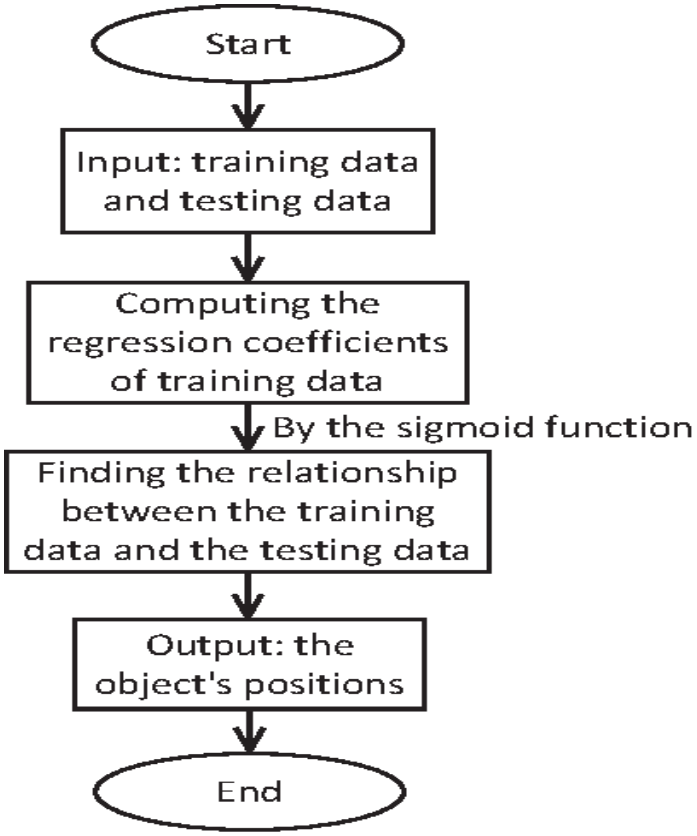


Fig. 6.7 Flowchart of Logistic Regression

6.4 Google Chrome Extension

Extensions are small software programs that customize the browsing experience. They enable users to tailor Chrome functionality and behaviour to individual needs or preferences. They are built on web technologies such as HTML, JavaScript, and CSS.

An extension must fulfil a single purpose that is narrowly defined and easy to understand. A single extension can include multiple components and a range of functionality, if everything contributes towards a common purpose.

User interfaces should be minimal and have intent. They can range from a simple icon, such as  the Google Mail Checker extension shown on the right, to overriding an entire page.

Extension files are zipped into a single .crx package that the user downloads and installs. This means extensions do not depend on content from the web, unlike ordinary web apps.

Extensions are distributed through the Chrome Developer Dashboard and published to the Chrome Web Store. For more information, see the store developer documentation.

The model of the algorithm that yields the maximum accuracy, i.e. the random forest algorithm is implemented on the extension. The features selected by random forest algorithm are: URL of Anchor, HTTPS and web traffic. The URL of Anchor is obtained by scraping the website opened. The HTTPS is obtained by acquiring the protocol of the opened website. The URL of the website is fed to https://www.alexa.com/siteinfo which provides the web traffic information. The chrome extension checks the rules of these features and compares them to the model. Based on the model, we can determine if the website is a phishing website or a legitimate website with an accuracy of 92.4%. If the website is a phishing website, the chrome extension would automatically raise an alert that informs the user that the website might be a phishing website.

Chapter 7

IMPLEMENTATION

Project implementation means carrying out the activities described in the work plan. It is the phase where visions and plans become reality.

7.1 Feature Selection Algorithms

Feature Selection is the process where you automatically or manually select those features which contribute most to your prediction variable or output in which you are interested in. Having irrelevant features in your data can decrease the accuracy of the models and make your model learn based on irrelevant features.

Benefits of Feature Selection:

- Reduces Over-fitting: Less redundant means less opportunity to make decisions based on noise.
- Improves Accuracy: Less mis-leading data means modelling accuracy improves.
- Reduces Training Time: Fewer data points reduce algorithm complexity and algorithms train faster.

7.1.1 Recursive Feature Elimination (RFE)

Recursive feature elimination (RFE) is a feature selection method that fits a model and removes the weakest feature (or features) until the specified number of features is reached. Features are ranked by the model's coefficients or feature-importance attributes, and by recursively eliminating a small number of features per loop, RFE attempts to eliminate dependencies and collinearity that may exist in the model.

RFE requires a specified number of features to keep, however it is often not known in advance how many features are valid. To find the optimal number of features cross-validation is used with RFE to score different feature subsets and select the best scoring collection of features. The RFECV visualizer plots the number of features in the model along with their cross-validated test score and variability and visualizes the selected number of features.

7.1.2 Sequential Feature Selector (SFS)

Automatic feature selection is an optimization technique that, given a set of features, attempts to select a subset of size that leads to the maximization of some criterion function. Feature selection algorithms are important to recognition and classification systems because, if a feature space with a large dimension is used, the performance of the classifier will decrease with respect to execution time and to recognition rate. The execution time increases with the number of features because of the measurement cost. The recognition rate can decrease because of redundant features and of the fact that small number of features can alleviate the course of dimensionality when the training samples set is limited, leading to overtraining. On the other hand, a reduction in the number of features may lead to a loss in the discrimination power and thereby lower the accuracy of the recognition system.

In order to determine the best feature subset for some criterion, some automatic feature selection algorithm can be applied to the complete feature space, varying the number of selected features from / to.

There are many different automatic feature selection methods. The most effective feature selection techniques are the sequential floating search methods (SFSM). There are two main categories of floating search methods: forward (SFFS) and backward (SFBS).

Basically, in the case of forward search (SFFS), the algorithm starts with a null feature set and, for each step, the best feature that satisfies some criterion function is included with the current feature set, i.e., one step of the sequential forward selection (SFS) is performed. The algorithm also verifies the possibility of improvement of the criterion if some feature

is excluded. In this case, the worst feature (concerning the criterion) is eliminated from the set, that is, it is performed one step of sequential backward selection (SBS). Therefore, the SFFS proceeds dynamically increasing and decreasing the number of features until the desired is reached.

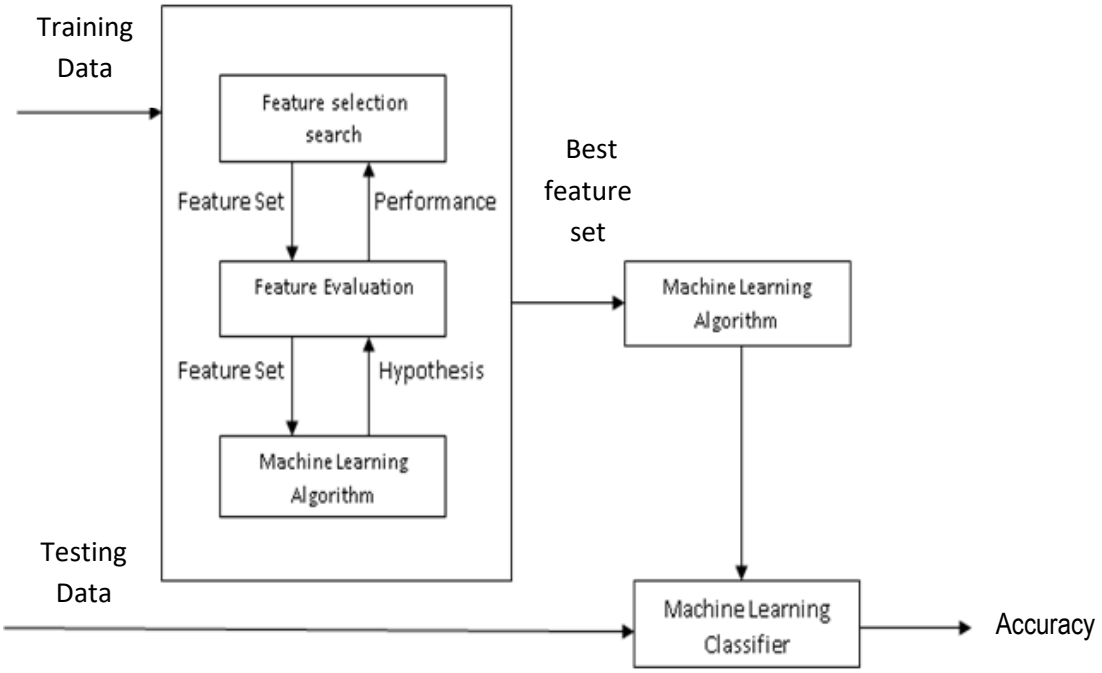


Fig. 7.1 The Feature Selection Approach used for predicting the phishing websites

7.2 Implementation Model

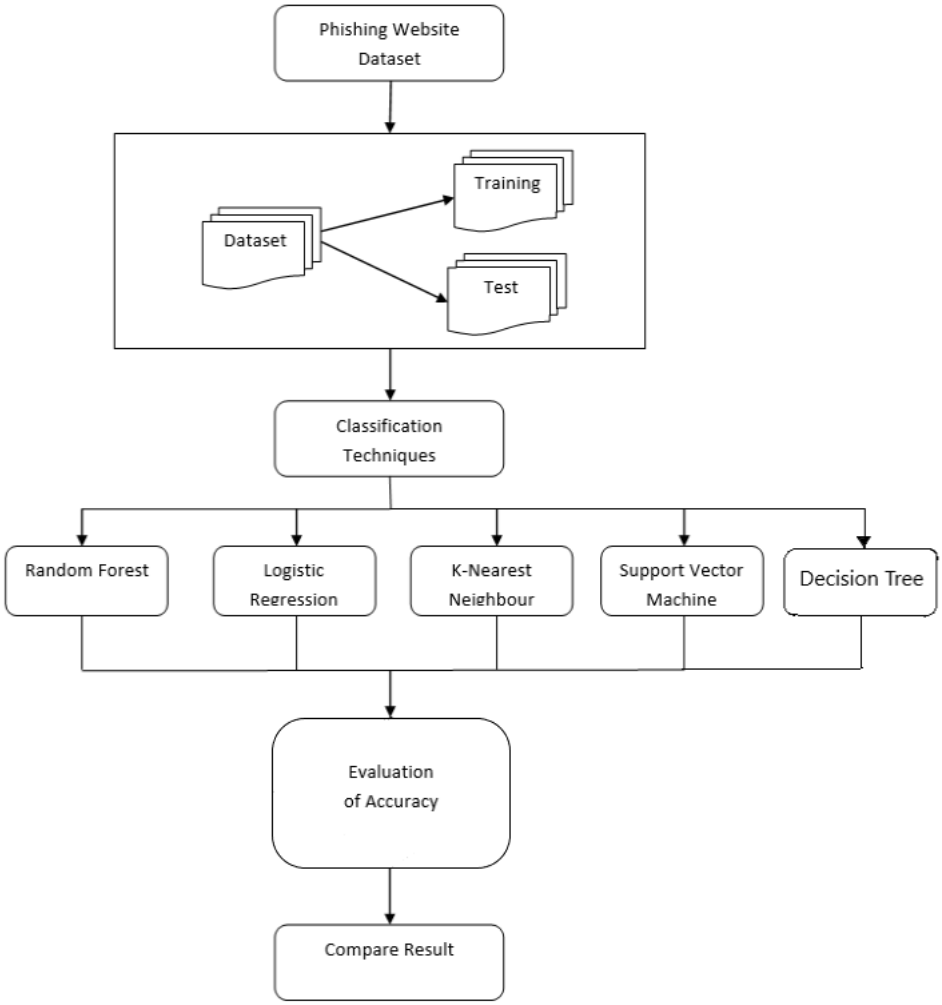


Fig. 7.2 Implementation of the proposed model

7.3 Implementation of the google chrome extension

The Phishing Alert is a tool allowing easy lookup of information relating to the sites you visit and providing protection against Phishing sites. It provides comprehensive site information and phishing protection when browsing the web. It is a powerful extension to detect phishing attacks in web sites and raise an alert when a phishing website is visited.

It provides a user-friendly interface by providing the following conveniences:

- website information can be directly accessed from within the extension.
- Site data refreshes if the current page redirects.

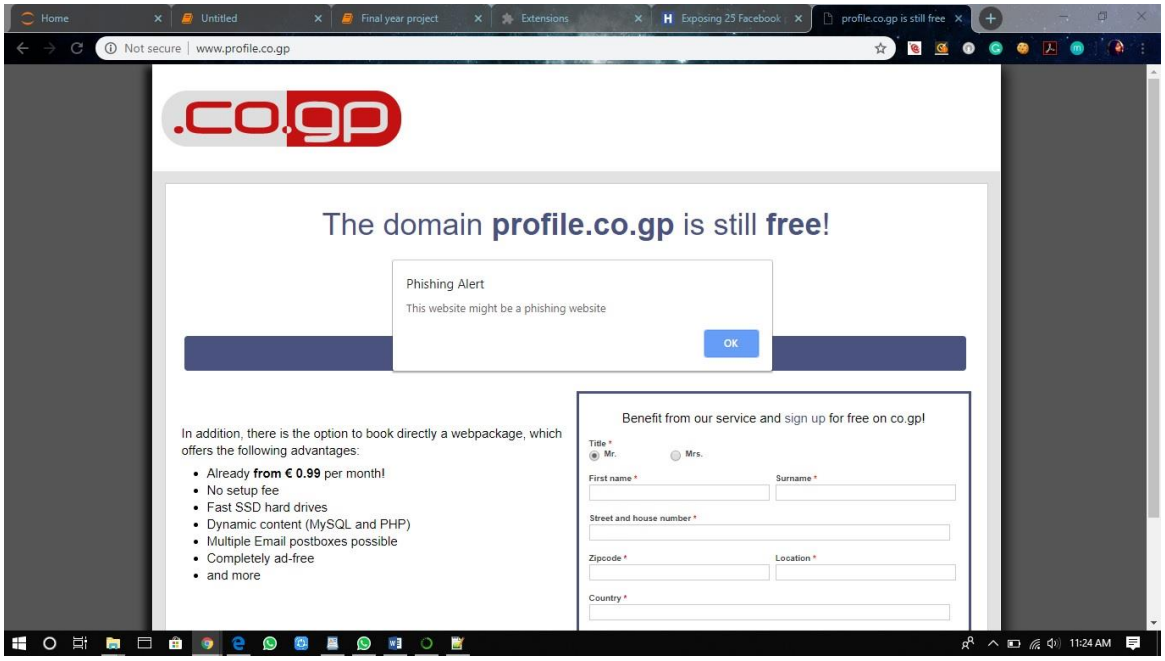


Fig. 7.3 Phishing Alert

7.4 Implementation Code

7.4.1 Algorithms.py

The below code is written in python script. It implements all the five algorithms i.e., SVM, DT, KNN, RF and LR along with RFE and SFS. Also, finds the accuracy of each of the above five mentioned algorithms along with their comparison.

```
from sklearn.svm import SVC  #Support Vector Machines(svm) & Classification(svc)
from sklearn.feature_selection import RFE #Recursive Feature Elimination
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn import tree
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
```

```
from sklearn import preprocessing, model_selection, svm, neighbors
from matplotlib.pyplot import figure
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import r2_score
from pandas import read_csv
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
sns.set(style="white")
sns.set(style="whitegrid", color_codes=True)

colors = ["#1f77b4", "#ff7f0e", "#2ca02c", "#d62728", "#8c564b"]

df = pd.read_csv('data.csv')
df.columns =
['having_IP_Address','URL_Length','Shortening_Service','having_At_Symbol','double_sl
ash_redirecting',
'Prefix_Suffix','having_Sub_Domain','SSLfinal_State','Domain_registration_length','Favi
con','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']

names = df.head()
X = df[df.columns[:-1]].values
y = df[['Result']].values

svc = SVC(kernel="linear", C=1)  #To convert n-dimentional data to linear data by
obtaining the dot product of the vectors
#Kernel describes the hyper-plane. 'rbf' can lead to overfitting (radial basis function)
# Penalty parameter C of the error term.
#It also controls the trade off between smooth decision boundary and classifying the
training points correctly.
rfe = RFE(estimator=svc, n_features_to_select=3, step=1)
# estimator assigns weights to features
#step corresponds to the number of features to remove at each iteration
#Fit the RFE model and then the underlying estimator on the selected features.
rfe.fit(X, y.ravel())

XX=[]
YY=[]
mm=sorted(zip(map(lambda x: round(x, 4), rfe.ranking_), names))
for i in mm:
```

```
    XX.append(i[0])
    YY.append(i[1])

plt.bar(YY[:3],XX[:3],align='center', alpha=0.5,color=colors)
plt.xlabel('Feature Selection')
plt.ylabel('RANK')
plt.title("Feature Selection BY SVM");
plt.show()

cols=[]
cols.append(mm[0][1])
cols.append(mm[1][1])
cols.append(mm[2][1])

sns.set(style='whitegrid', context='notebook')
sns.pairplot(df[cols], height=1.5);
plt.show()

cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale = 1.5)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',
annot_kws={'size': 15}, yticklabels=cols, xticklabels=cols)
plt.show()

X_new = rfe.transform(X)  #Reduce X to the selected features.
X_train, X_test, y_train, y_test = train_test_split(X_new, y, test_size = 0.25)
svc.fit(X_train, y_train.ravel())
ysvc = svc.predict(X_test)

mse=[]
mae=[]

rsq=[]
rmse=[]
acy=[]

print("MSE VALUE FOR SVM IS %f "  % mean_squared_error(y_test,ysvc))
print("MAE VALUE FOR SVM IS %f "  % mean_absolute_error(y_test,ysvc))
print("R-SQUARED VALUE FOR SVM IS %f "  % r2_score(y_test,ysvc))
rms = np.sqrt(mean_squared_error(y_test,ysvc))
print("RMSE VALUE FOR SVM IS %f "  % rms)
ac=accuracy_score(y_test,ysvc) * 100
print ("ACCURACY VALUE SVM IS %f" % ac)

mse.append(mean_squared_error(y_test,ysvc))
mae.append(mean_absolute_error(y_test,ysvc))
rsq.append(r2_score(y_test,ysvc))
rmse.append(rms)
acy.append(ac)
```

```
lr = LogisticRegression(solver = 'lbfgs')
# create the RFE model and select 3 attributes
rfe = RFE(lr, 3)
rfe = rfe.fit(X,y.ravel())

mm=sorted(zip(map(lambda x: round(x, 4), rfe.ranking_), names))

a1=mm[0]
a2=mm[1]
a3=mm[2]


XX=[]
YY=[]
XX.append(a1[1])
YY.append(1)
XX.append(a2[1])
YY.append(1)
XX.append(a3[1])
YY.append(1)

#Barplot for the dependent variable
fig = plt.figure(0)
plt.bar(XX,YY,align='center', alpha=0.5,color=colors)
plt.xlabel('Feature Selection')
plt.ylabel('RANK')
plt.title("Feature Selection BY LogisticRegression");

plt.show()

cols=[]

cols.append(a1[1])
cols.append(a2[1])
cols.append(a3[1])

sns.set(style='whitegrid', context='notebook')
sns.pairplot(df[cols], height=1.5);
plt.show()

cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale = 1.5)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',
annot_kws={'size': 15}, yticklabels=cols, xticklabels=cols)
plt.show()

X_new = rfe.transform(X)
```

```
X_train, X_test, y_train, y_test = model_selection.train_test_split(X_new, y, test_size =
0.25)
lr.fit(X_train, y_train.ravel())
ylr = lr.predict(X_test)

print("MSE VALUE FOR LogisticRegression IS %f "  %
mean_squared_error(y_test,ylr))
print("MAE VALUE FOR LogisticRegression IS %f "  %
mean_absolute_error(y_test,ylr))
print("R-SQUARED VALUE FOR LogisticRegression IS %f "  % r2_score(y_test,ylr))
rms = np.sqrt(mean_squared_error(y_test,ylr))
print("RMSE VALUE FOR LogisticRegression IS %f "  % rms)
ac=accuracy_score(y_test,ylr) * 100
print ("ACCURACY VALUE LogisticRegression IS %f" % ac)
mse.append(mean_squared_error(y_test,ylr))
mae.append(mean_absolute_error(y_test,ylr))
rsq.append(r2_score(y_test,ylr))
rmse.append(rms)
acy.append(ac)

dtree = tree.DecisionTreeClassifier()
rfe = RFE(estimator=dtree, n_features_to_select=3)
rfe.fit(X, y.ravel())

mm=sorted(zip(map(lambda x: round(x, 4), rfe.ranking_), names))
a1=mm[0]
a2=mm[1]
a3=mm[2]


XX=[]
YY=[]

XX.append(a1[1])
YY.append(1)
XX.append(a2[1])
YY.append(1)
XX.append(a3[1])
YY.append(1)

#Barplot for the dependent variable
fig = plt.figure(0)
plt.bar(XX,YY,align='center', alpha=0.5,color=colors)
plt.xlabel('Feature Selection')
plt.ylabel('RANK')
plt.title("Feature Selection BY DecisionTree");
plt.show()

cols=[]
```

```
cols.append(a1[1])
cols.append(a2[1])
cols.append(a3[1])

sns.set(style='whitegrid', context='notebook')
sns.pairplot(df[cols], height=1.5);
plt.show()

cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale = 1.5)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',
annot_kws={'size': 15}, yticklabels=cols, xticklabels=cols)
plt.show()


X_new = rfe.transform(X)
X_train, X_test, y_train, y_test = model_selection.train_test_split(X_new, y, test_size =
0.25)
dtree.fit(X_train, y_train)
ydtree = dtree.predict(X_test)
print("MSE VALUE FOR DecisionTree IS %f "  % mean_squared_error(y_test,ydtree))
print("MAE VALUE FOR DecisionTree IS %f "  % mean_absolute_error(y_test,ydtree))
print("R-SQUARED VALUE FOR DecisionTree IS %f "  % r2_score(y_test,ydtree))
rms = np.sqrt(mean_squared_error(y_test,ydtree))
print("RMSE VALUE FOR DecisionTree IS %f "  % rms)
ac=accuracy_score(y_test,ydtree) * 100
print ("ACCURACY VALUE DecisionTree IS %f" % ac)
mse.append(mean_squared_error(y_test,ydtree))
mae.append(mean_absolute_error(y_test,ydtree))

rsq.append(r2_score(y_test,ydtree))
rmse.append(rms)
acy.append(ac)

rf = RandomForestClassifier(n_estimators = 10)
rfe = RFE(estimator=rf, n_features_to_select=3)
rfe.fit(X, y.ravel())

mm=sorted(zip(map(lambda x: round(x, 4), rfe.ranking_), names))
a1=mm[0]
a2=mm[1]
a3=mm[2]

XX=[]
YY=[]
XX.append(a1[1])
YY.append(1)
XX.append(a2[1])
YY.append(1)
```

```
XX.append(a3[1])
YY.append(1)

#Barplot for the dependent variable
fig = plt.figure(0)
plt.bar(XX,YY,align='center', alpha=0.5,color=colors)
plt.xlabel('Feature Selection')
plt.ylabel('RANK')
plt.title("Feature Selection BY RandomForest");
plt.show()

cols=[]
cols.append(a1[1])
cols.append(a2[1])
cols.append(a3[1])

sns.set(style='whitegrid', context='notebook')
sns.pairplot(df[cols], height=1.5);
plt.show()

cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale = 1.5)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',
annot_kws={'size': 15}, yticklabels=cols, xticklabels=cols)
plt.show()

X_new = rfe.transform(X)
X_train, X_test, y_train, y_test = model_selection.train_test_split(X_new, y, test_size =
0.25)

rf.fit(X_train, y_train.ravel())
yrf = rf.predict(X_test)

print("MSE VALUE FOR Random Forest IS %f "  % mean_squared_error(y_test,yrf))
print("MAE VALUE FOR Random Forest IS %f "  % mean_absolute_error(y_test,yrf))
print("R-SQUARED VALUE FOR Random Forest IS %f "  % r2_score(y_test,yrf))
rms = np.sqrt(mean_squared_error(y_test,yrf))
print("RMSE VALUE FOR Random Forest IS %f "  % rms)
ac=accuracy_score(y_test,yrf) * 100
print ("ACCURACY VALUE Random Forest IS %f" % ac)
mse.append(mean_squared_error(y_test,yrf))
mae.append(mean_absolute_error(y_test,yrf))
rsq.append(r2_score(y_test,yrf))
rmse.append(rms)
acy.append(ac)

knn = KNeighborsClassifier(n_neighbors=4)
sfs1 =
SFS(knn,k_features=3,forward=True,floating=False,verbose=2,scoring='accuracy',cv=0)
```

```
sfs1 = sfs1.fit(X, y.ravel())
print("-----------------")
print(sfs1.k_feature_idx_)
XX=[]
YY=[]
for kk in sfs1.k_feature_idx_:
print(kk)
print(df.columns[kk])
XX.append(df.columns[kk])
YY.append(1)
print(sfs1.k_score_)

#Barplot for the dependent variable
fig = plt.figure(0)
plt.bar(XX,YY,align='center', alpha=0.5,color=colors)
plt.xlabel('Feature Selection')
plt.ylabel('RANK')
plt.title("Feature Selection BY KNN");
plt.show()

cols=[]
cols.append(a1[1])
cols.append(a2[1])
cols.append(a3[1])

sns.set(style='whitegrid', context='notebook')

sns.pairplot(df[cols], height=1.5);
plt.show()

cm = np.corrcoef(df[cols].values.T)
sns.set(font_scale = 1.5)
hm = sns.heatmap(cm, cbar=True, annot=True, square=True, fmt='.2f',
annot_kws={'size': 15}, yticklabels=cols, xticklabels=cols)
plt.show()


X_new = rfe.transform(X)
X_train, X_test, y_train, y_test = model_selection.train_test_split(X_new, y, test_size =
0.25)
knn.fit(X_train, y_train.ravel())
yknn = knn.predict(X_test)
print(yknn)
accuracy = 100.0 * accuracy_score(y_test, yknn)
print ("The accuracy is: " + str(accuracy))

print("MSE VALUE FOR KNN IS %f "  % mean_squared_error(y_test,yknn))
print("MAE VALUE FOR KNN IS %f "  % mean_absolute_error(y_test,yknn))
print("R-SQUARED VALUE FOR KNN IS %f "  % r2_score(y_test,yknn))
```

```
rms = np.sqrt(mean_squared_error(y_test,yknn))
print("RMSE VALUE FOR KNN IS %f "  % rms)
ac=accuracy_score(y_test,yknn) * 100
print ("ACCURACY VALUE KNN IS %f" % ac)
mse.append(mean_squared_error(y_test,yknn))
mae.append(mean_absolute_error(y_test,yknn))
rsq.append(r2_score(y_test,yknn))
rmse.append(rms)
acy.append(ac)

al = ['SVM','KNN','RF','LR','DT']
l = np.arange(len(al))

plt.bar(l,mse,align='center', alpha=0.5,color=colors)
plt.xticks(l, al)
plt.xlabel('Algorithm')
plt.ylabel('MSE')
plt.title("MSE Value");
plt.show()

plt.bar(l,mae,align='center', alpha=0.5,color=colors)
plt.xticks(l, al)
plt.xlabel('Algorithm')
plt.ylabel('MAE')
plt.title('MAE Value')
plt.show()

plt.bar(l,rsq,align='center', alpha=0.5,color=colors)
plt.xticks(l, al)

plt.xlabel('Algorithm')
plt.ylabel('R-SQUARED')
plt.title('R-SQUARED Value')
plt.show()

plt.bar(l, rmse, align='center', alpha=0.5,color=colors)
plt.xticks(l, al)
plt.xlabel('Algorithm')
plt.ylabel('RMSE')
plt.title('RMSE Value')
plt.show()

plt.bar(l, acy,align='center', alpha=0.5,color=colors)
plt.xticks(l, al)
plt.xlabel('Algorithm')
plt.ylabel('Accuracy')
plt.title('Accuracy Value')
plt.show()
```

7.4.2 contentScript.js

The below code is written in JavaScript.

```
var a = 0;
$("a").each(function(){
  if((this.href).matches(/javascript:void(0)/g) || (this.target).matches(/_blank/g)){
  a=1;
  }
});
var b;
if(window.location.protocol != "https:"){
b=a+1;
}
chrome.runtime.sendMessage(b);
```

7.4.3 background.js

The below code is written in JavaScript.

```
var url,b = 0,rank = new Array(),gRank;
chrome.runtime.onMessage.addListener(function(response, sender, sendResponse){
url = sender.url;
url = "https://www.alexa.com/siteinfo/"+url;
$.get(url,function(data){
$(data).find("strong").each(function(){
if((this.innerText).match(/[0-9]+/g) != null){
rank.push(this.innerText);
}
});
if(rank[0] != undefined){
var gRank = (parseInt((rank[0]).replace(/,/g,"")));
}
$(data).find("strong").each(function(){
var str = this.innerText;
if(str.match(/We don't have enough data to rank this website./g)){
```

```
b = 1;

}

});

if((( b == 1 || gRank > 100000 ) && (response == 1 || response == 2))){

chrome.runtime.sendMessage([response,b,gRank]);

alert("This website might be a phishing website");

}

});

});
```

7.4.4 background.html

The below code is written in html.

```
<!DOCTYPE html>
<html>
<head>
<script src = "jquery.js"></script>
<script src = "background.js"></script>
</head>
<body>
</body>
</html>
```

7.4.5 manifest.json

The below code is written in JSON script.

```
{
  "name": "Phishing Alert",
  "description" : "Base Level Extension",
  "version": "1.0",
  "permissions": ["tabs"],
  "browser_action": {
    "default_popup": "hello.html",
    "default_icon": "hello_extensions.png"
  },
  "icons" : {
"64" : "hello_extensions1.png"
  },
```

```
  "manifest_version": 2,
  "commands": {
    "_execute_browser_action": {
      "suggested_key": {
        "default": "Ctrl+Shift+F",
        "mac": "MacCtrl+Shift+F"
      },
      "description": "Opens hello.html"
    }
  },
   "background": {
    "matches": ["<all_urls>"],
    "page": "background.html"
  },
  "content_scripts": [
   {
     "matches": ["<all_urls>"],
"run_at": "document_start",
     "js": ["jquery.js","contentScript.js"]
   }
 ]
}
```

Chapter 8

DISCUSSION AND SCREENSHOTS

8.1 Heatmaps

A heat map (or heatmap) is a graphical representation of data where the individual values contained in a matrix are represented as colours. It is data analysis software that uses colour the way a bar graph uses height and width: as a data visualization tool.

Here, it shows how the three selected features for that particular algorithm are correlated with each other. In these heatmaps, the white colour indicates that the two features considered have high correlation and the darker colour indicates that the two features considered have least correlation.

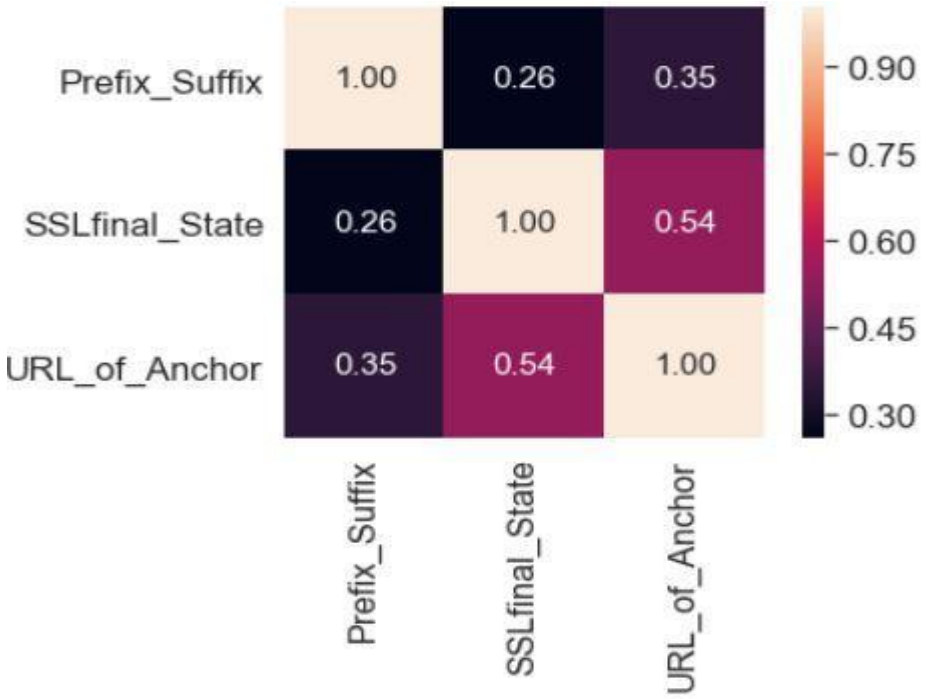


Fig. 8.1 Heatmap of Support Vector Machine Algorithm

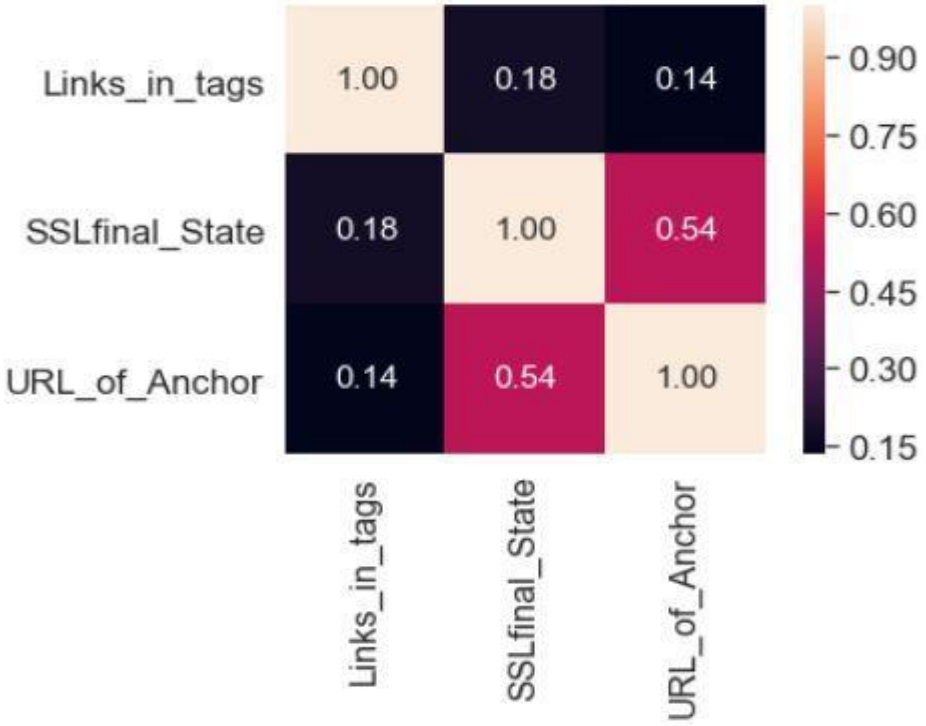


Fig. 8.2 Heatmap of Decision Tree Algorithm

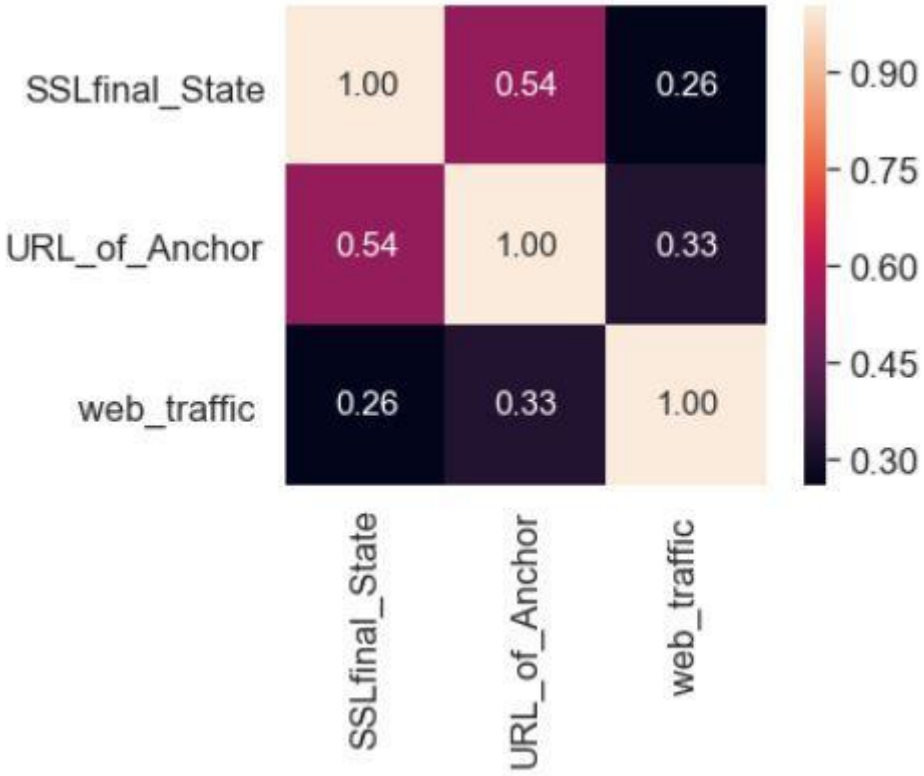


Fig. 8.3 Heatmap of K-Nearest Neighbours Algorithm

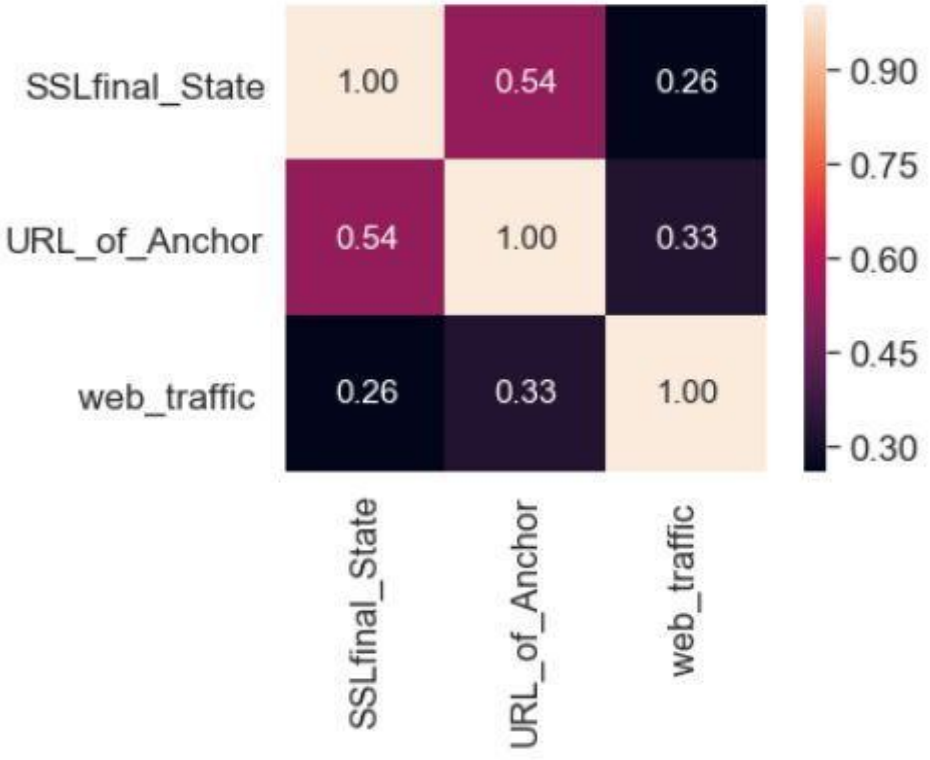


Fig. 8.4 Heatmap of Random Forest Algorithm

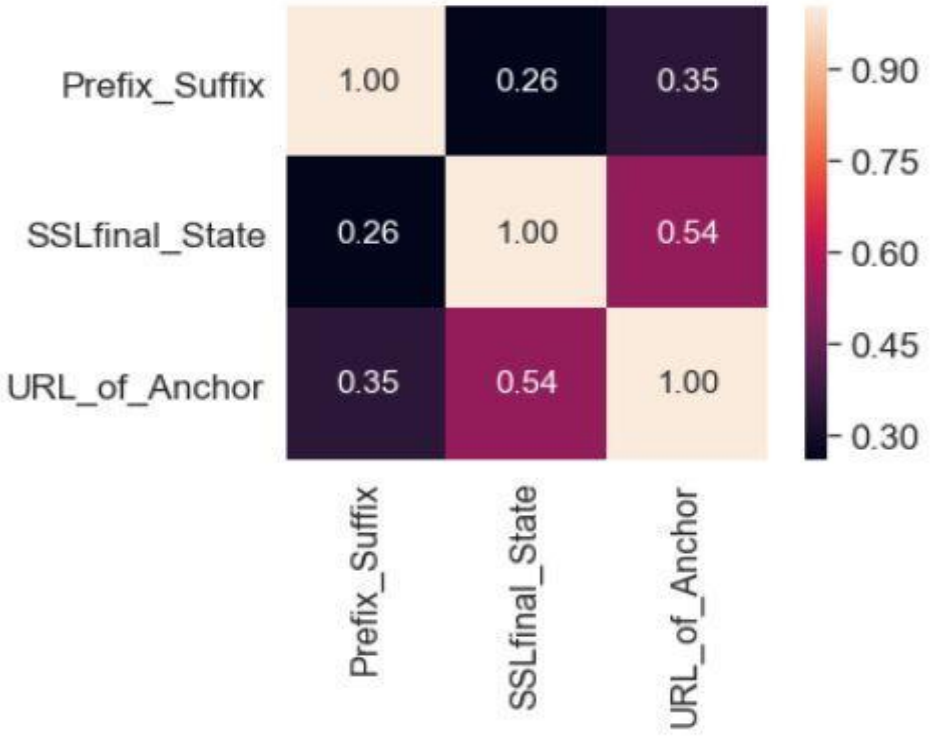


Fig. 8.5 Heatmap of Logistic Regression Algorithm

8.2 Comparison Graphs of various Metrics

8.2.1 Root-mean-square error (RMSE)

Root-mean-square error (RMSE) is a frequently used measure of the differences between values predicted by a model or an estimator and the values actually observed.

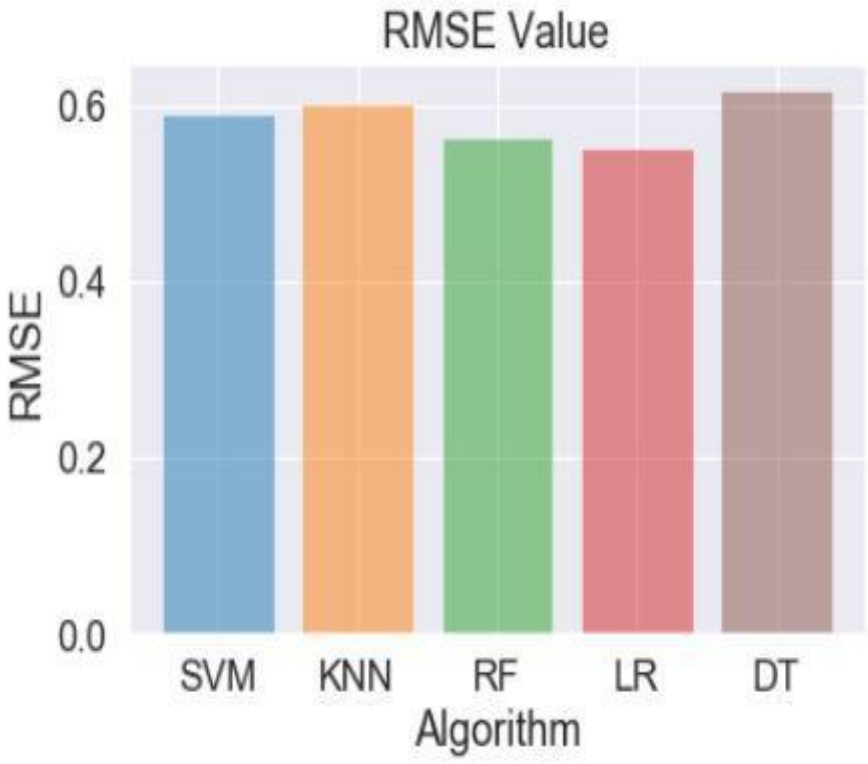


Fig. 8.6 RMSE Plot

8.2.2 R-squared value (R^2)

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination, or the coefficient of multiple determination for multiple regression.

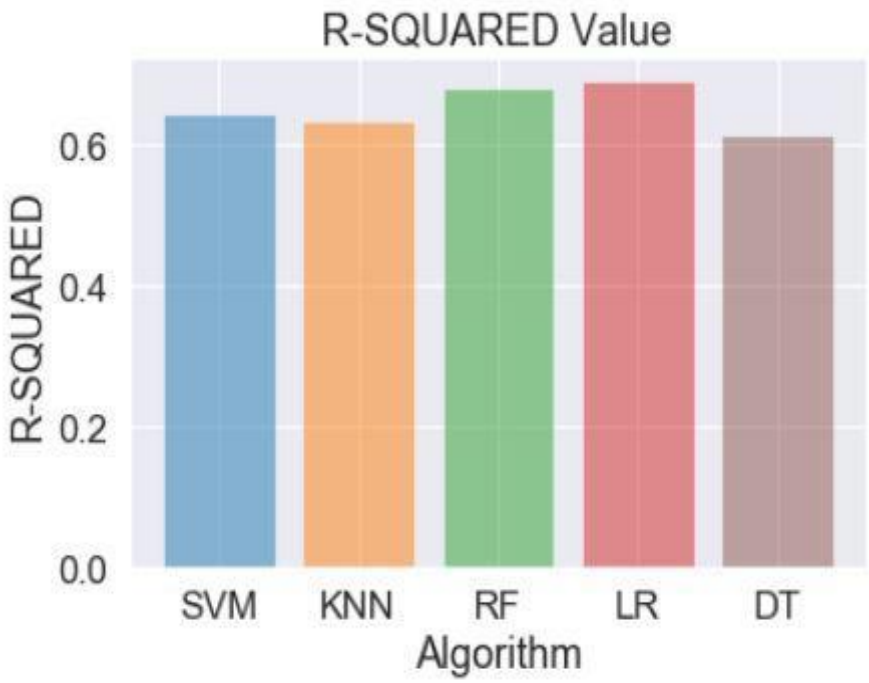


Fig. 8.7 R squared value Plot

8.2.3 Mean Absolute Error (MAE)

The Mean Absolute Error (or MAE) is the sum of the absolute differences between predictions and actual values. It gives an idea of how wrong the predictions were. The measure gives an idea of the magnitude of the error, but no idea of the direction (e.g. over or under predicting).

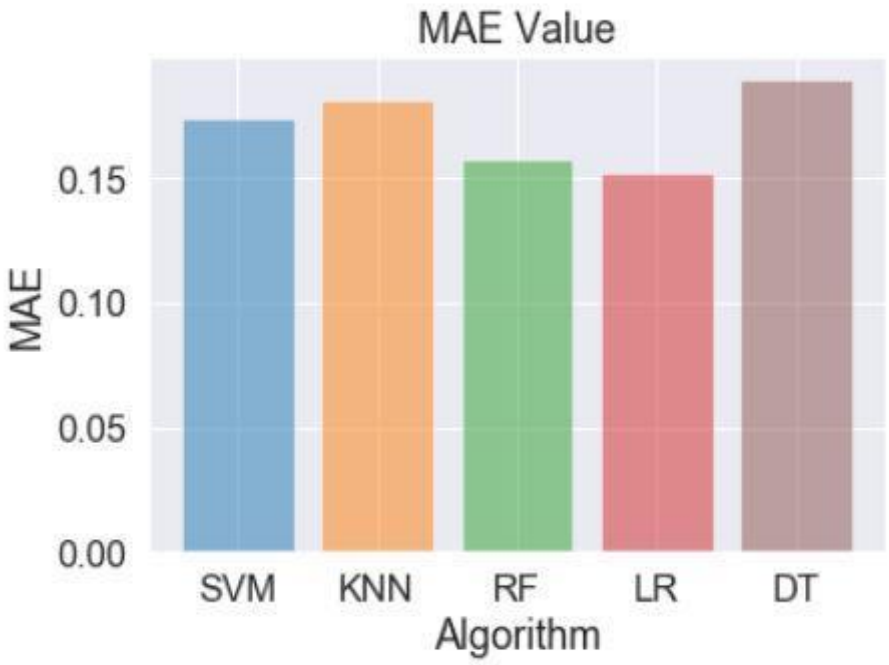


Fig. 8.8 MAE Plot

8.2.4 The Mean Squared Error (MSE)

The Mean Squared Error (or MSE) is much like the mean absolute error in that it provides a gross idea of the magnitude of error.

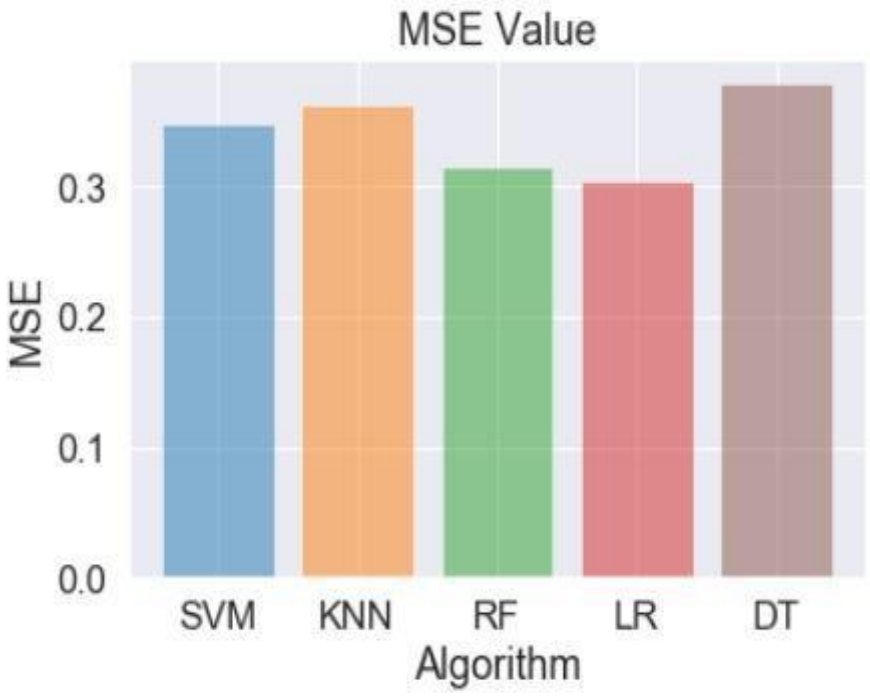


Fig. 8.9 MSE Plot

Chapter 9

RESULT AND ANALYSIS

Implementation of all the five machine learning algorithms on the given dataset for phishing website detection shows that Random Forest model outperforms other models. The accuracy of Random Forest is high compared to other machine learning algorithms.

Algorithms	MSE	MAE	R-Squared value	RMSE	Accuracy
SVM	0.348770	0.174385	0.645547	0.590567	91.280753
Logistic Regression	0.363242	0.181621	0.632992	0.602695	90.18958
Decision Tree	0.315485	0.157742	0.681244	0.561680	92.112880
Random Forest	0.303907	0.151954	0.690486	0.551278	92.402315
KNN	0.379161	0.189580	0.613997	0.615760	90.520984

Table 1: Evaluation metrics of various algorithms

The below plot represents accuracy of machine learning algorithms SVM, Decision tree, KNN, Random Forest and Logistic Regression for phishing website detection.

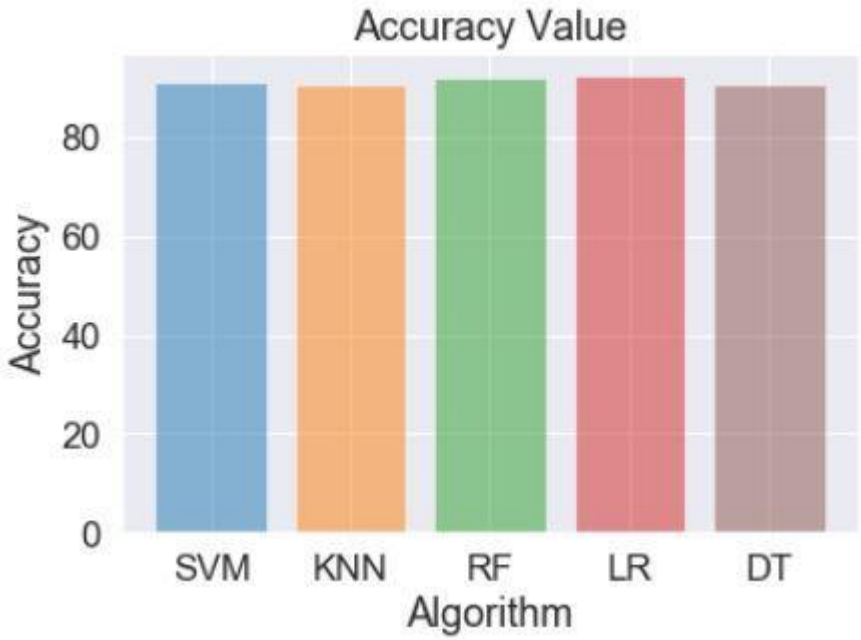


Fig. 9.1 Accuracy plot

Chapter 10

CONCLUSION

Phishing is a cybercrime procedure utilizing both social building and specialized deception to take individual sensitive data. Besides, Phishing is considered as another extensive type of fraud. Experimentations against recent dependable phishing data sets utilizing different classification algorithms have been performed which received different learning methods. The base of the experiments is the accuracy measure. The aim of this research work is to predict whether website that is opened while surfing the web is a phishing website or not. As future work, we might use this model to test other Phishing dataset with a larger size and use more features to improve its accuracy.