Detecting Phishing Attacks Using Natural Language Processing and Machine Learning

B. Jeevitha[1], G. Samhitha[2,] T. Pranay[3], Mr. M.Rajkumar[4], Dr. P. Srinivasa Rao[5],
UG Scholar[1,2,3], Assistant professor[4,] Professor & HoD[5],
Department of Computer Science and Engineering[1,2,3,4,5],
J.B. Institute OF Engineering & Technology[1,2,3,4,5],
Moinabad, R.R. District, Hyderabad, Telangana, India.

Abstract: Malicious websites contribute significantly to the expansion of online criminal activity and stifle the development of Web services. As a result, there has been a tremendous push to build a comprehensive solution to prevent users from accessing such websites. We propose a learning-based method for categorising Web sites into three categories: benign, spam, and malicious. Without accessing the content of Web sites, our technique solely analyses the Uniform Resource Locator (URL). As a result, run-time latency is eliminated, as is the risk of users being exposed to browser-based vulnerabilities. In comparison to boycotting administration, our method achieves greater execution on all-inclusive statement and inclusion by employing learning calculations.

Keyword: *Machine Learning, legitimate, URL*

I. INTRODUCTION

While the Internet has provided tremendous convenience for many people in managing their finances and business activities, it also provides opportunities for extortionists to target a large number of people at a low cost. Clients, rather than equipment or programming, are under the hands of fraudsters. frameworks, where barriers to inventive trade-off have virtually increased. Phishing is one of the most often practised Internet impersonations. It revolves around the theft of sensitive personal information such as passwords and MasterCard details. Phishing attacks are divided into two types:

- Attempts to dupe exploited people into disclosing their insider information by posing as trustworthy sources with a real need for such information.
- attempts to gain privileged information by infecting the computers of unwitting victims with malware.

The infection and malware network is investigating the malware used in phishing attacks, but it isn't being addressed right now. The examination focal point of this postulation is phishing assaults that continue by misdirecting customers, and the word 'phishing assault' will be used to refer to this type of attack.

II. OBJECTIVE

The main objective of this paper is to detect the Begin, Malicious and Malware URLs with the use of machine learning.

MOTIVATION

The reason behind this system is to take precautions to prevent users from these harmful sites. It will make people conscious in addition to building strong security mechanisms which are able to detect and prevent phishing URL's from reaching the user.

III. PROBLEM STATEMENT

URLs of the websites are separated into 3 classes:

- Benign: Safe websites with normal services
- Spam: Website performs the act of attempting to flood the user with advertising or sites such as fake surveys and online dating etc.
- Malware: Website created by attackers to disrupt computer operation, gather sensitive information, or gain access to private computer systems.

Existing System:

A poorly structured NN model may cause the model to under fit the training dataset. On the other hand, exaggeration in restructuring the system to suit every single item in the training dataset may cause the system to be over fitted. One possible solution to avoid the Overfitting problem is by restructuring the NN model in terms of tuning some parameters, adding new neurons to the hidden layer or sometimes adding a new layer to the network. A NN with a small number of hidden neurons may not have a satisfactory representational power to model the complexity and diversity inherent in the data. On the other hand, networks with too many hidden neurons could overfit the data. However, at a certain stage the model can no longer be improved, therefore, the structuring process should be terminated. Hence, an acceptable error rate should be specified when creating any NN model, which itself is considered a problem since it is difficult to determine the acceptable error rate a priori. For instance, the model designer may set the acceptable error rate to a value that is unreachable which causes the model to stick in local minima or sometimes the model designer may set the acceptable error rate to a value that can further be improved.

Disadvantage:

1. It will take time to load all the dataset.

2. Process is not accuracy.
3. It will analyse slowly.

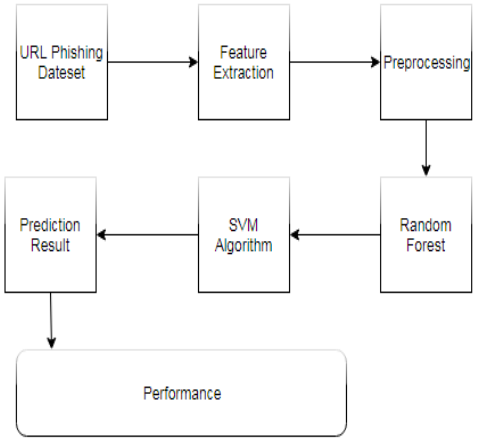
IV. PROPOSED SYSTEM

Lexical features are based on the observation that the URLs of many illegal sites look different, compared with legitimate sites. Analysing lexical features enables us to capture the property for classification purposes. We first distinguish the two parts of a URL: the host name and the path, from which we extract bag-of-words (strings delimited by '/', '?', '.', '=', '-' and ''). We find that phishing site wants to have longer URL, more levels (delimited by spot), more tokens in area and way, longer token. Plus, phishing and malware sites could profess to be a kind one by containing mainstream brand names as tokens other than those in second-level area. Considering phishing sites and malware sites may utilize IP address straightforwardly in order to cover the suspicious URL, which is uncommon in generous case. Likewise, phishing URLs are found to contain a few intriguing word tokens (affirm, account, banking, secure, ebayisapi, webscr, login, signin), we check the nearness of these security touchy words and remember the paired an incentive for our highlights. Instinctively, malignant locales are in every case less famous than kind ones. Consequently, site ubiquity can be considered as a significant element. Traffic rank component is procured from Alexa.com. Host-put together highlights are based with respect to the perception that pernicious destinations are constantly enrolled in less trustworthy facilitating focuses or districts.

Advantage:

1. All of URLs in the dataset are labelled.
2. We used two supervised learning algorithms random forest and support vector machine to train using scikit-learn library.

System Architecture



METHODOLOGY

- **DATA COLLECTION**
- **DATA PRE-PROCESSING**
- **FEATURE EXTRATION**
- **EVALUATION MODEL**

DATA COLLECTION:

Data used in this paper is a set of records. This step is concerned with selecting the subset of all available data that you will be working with. ML problems start with data preferably, lots of data (examples or observations) for which you already know the target answer. Data for which you already know the target answer is called *labelled data*.

DATA PRE-PROCESSING

Organize your selected data by formatting, cleaning and sampling from it.

Three common data pre-processing steps are:

1. Formatting
2. Cleaning
3. Sampling

Formatting: The information you have chosen may not be in an arrangement that is reasonable for you to work with. The information might be in a social database and you might want it in a level record, or the information might be in an exclusive document organization and you might want it in a social database or a book document.

Cleaning: Cleaning information is the expulsion or fixing of missing information. There might be information examples that are inadequate and don't convey the information you trust you have to address the issue. These occasions may should be expelled. Also, there might be touchy data in a portion of the traits and these ascribes may should be anonym zed or expelled from the information.

Testing: There might be definitely more chosen information accessible than you have to work with. More information can bring about any longer running occasions for calculations and bigger computational and memory prerequisites. You can take a littler agent test of the chose information that might be a lot quicker for investigating and prototyping arrangements before considering the entire dataset. Next thing is to do Feature extraction is an attribute extension we created more columns from URL's. Finally, our models are trained using Classifier algorithm. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data.

FEATURE EXTRATION

Next thing is to do Feature extraction is an attribute extension we created more columns from URL's. Finally, our models are trained using Classifier algorithm. We use the labelled dataset gathered. The rest of our labelled data will be used to evaluate the models. Some machine learning algorithms were used to classify pre-processed data. The chosen classifiers were Random forest.

EVALUATION MODEL

Model Evaluation is an indispensable piece of the model advancement process. It assists with finding the best model that speaks to our information and how well the picked model will function later on.

To maintain a strategic distance from over fitting, the two techniques utilize a test set (not seen by the model) to assess model execution. Execution of every order model is assessed base on its arrived at the midpoint of. The outcome will be in the envisioned structure. Portrayal of ordered information as charts. Exactness is characterized as the level of right forecasts for the test information. It very well may be determined effectively by isolating the quantity of right forecasts by the quantity of all out expectations. We anticipate the exactness over genuine and anticipated yield and compute precision as –

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP}$$

ALGORITHM

1) Randomly initialize populations p

2) Determine fitness of population

3) Until convergence repeat:

a) Select parents from population

b) Crossover and generate new population

c) Perform mutation on new population

d) Calculate fitness for new population

- First, we need to gather a large set of data, which has apt attributes and which are related to URL link study.
- Open Anaconda and Jupyter Notebook to write the code.
- Upload the dataset into the Jupyter to use in the program
- Preprocess the data
- Convert Categorical values to numerical values
- Replace numerical missing values with Mean
- Replace Categorical missing values with Mode.
- Then, we will split the dataset into training and test sample.
- Construct Random Forest Model
- Construct SVM model
- Construct Genetic Algorithm Model
- Check the accuracy of the constructed models using the confusion matrix
- The model is deployed into the pickle created.

Genetic Algorithm

Simulating the process of natural selection, reproduction and mutation, the genetic algorithms can produce high-quality solutions for various problems including search and optimization. By the effective use of the Theory of Evolution genetic algorithms are able to surmount problems faced by traditional algorithms. According to Darwin's theory of evolution, an evolution maintains a population of individuals that vary from each other (variation). Those who are better adapted to their environment have a greater chance of surviving, breeding, and passing their traits to the next generation (survival of the fittest).

Chromosome/Individual

A chromosome is a collection of genes. For example, a chromosome can be represented as a binary string where each bit is a gene. Genetic Algorithms are search algorithms inspired by Darwin's Theory of Evolution in nature.

By simulating the process of natural selection, reproduction and mutation, the genetic algorithms can produce high-quality solutions for various problems including search and optimization.

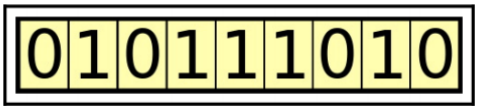
By the effective use of the Theory of Evolution genetic algorithms are able to surmount problems faced by traditional algorithms.

According to Darwin's theory of evolution, an evolution maintains a population of individuals that vary from each other (variation). Those who are better adapted to their

environment have a greater chance of surviving, breeding, and passing their traits to the next generation (survival of the fittest).

Chromosome/Individual

A chromosome is a collection of genes. For example, a chromosome can be represented as a binary string where each bit is a gene.



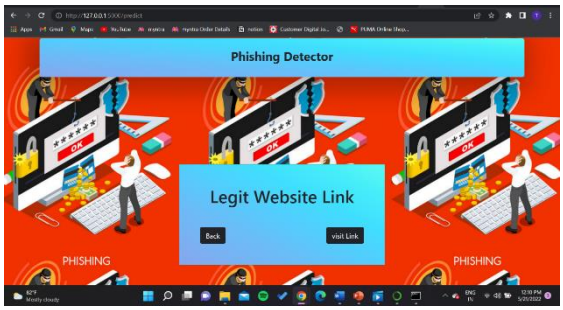
Simple binary-coded chromosome

Population

Since an individual is represented as a chromosome, a population is a collection of such chromosomes.

V. RESULTS

Open main.py using Anaconda terminal in Environment. We will get the link of the local host. Copy the link and paste it in the browser. We will get the screen shown below. To check the URL is a phishing URL or not, copy the link and pate the URL in the given box. Select the Random Forest algorithm to give best results.



If the entered URL is safe and doesn't cause any harm then it shows as "Legit Website Link" else "Fake link".

VI. CONCLUSION

In this paper, we describe our large-scale system for automatically classifying phishing pages which maintains a false positive rate below 0.1%. Our classification system examines millions of potential phishing pages daily in a fraction of the time of a manual review process. By automatically updating our blacklist with our classifier, we minimize the amount of time that phishing pages can remain active before we protect our users from them. Even with a perfect classifier and a robust system, we recognize that our blacklist approach keeps us perpetually a step behind the phishers. We can only identify a phishing URL and normal URL using machine learning algorithm. Result we got in terms of accuracy metric.

VII. REFERENCES

[1]. G. Aaron and R. Rasmussen, "Global phishing survey: Trends and domain name use in 2016," 2016.
[2]. B. Gupta, A. Tewari, A. K. Jain, and D. P. Agrawal, "Fighting against phishing attacks: state of the art and future challenges," Neural Computing and Applications, vol. 28, no. 12, pp. 3629–3654, 2017.
[3]. A. Aleroud and L. Zhou, "Phishing environments, techniques, survey,"countermeasures: Aand Security Computers & , vol. 68, pp. 160 – 196, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167404817300810 G. Aaron and R. Rasmussen, "Phishing activity trends report: 4th
[4]. quarter 2016," 2014. R. Verma, N. Shashidhar, and N. Hossain, "Detecting phishing
[5]. emails the natural language way," in Computer Security–ESORICS 2012. Springer, 2012, pp. 824–841. M. Khonji, Y. Iraqi, and A. Jones, "Phishing detection: a literature
[6]. survey," IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2091–2121, 2013. G. Park and J. M. Taylor, "Using syntactic features for phishing
[7]. detection," arXiv preprint arXiv:1506.00037, 2015. R. Dazeley, J. L. Yearwood, B. H. Kang, and A. V. Kelarev