RELATIVE STUDY OF MACHINE LEARNING AND DEEP LEARNING ALGORITHMS FOR URL-BASED PHISHING IDENTIFICATION

Raman Kumar[1], M Vishal[2], Yogesh Singh[3], Ayush Singh Chouhan[4]

[14]Student, School of Computer Science and Engineering, Lovely Professional University, India

Abstract - *Phishing costs billions of dollars annually for Internet users. It is one of the most significant challenges to cyber security. Phishers employ phishing tools, manipulated e-mails, to get and misuse sensitive information like personal and financial information such as credit and debit card details, login credentials like usernames, passwords and even can access their accurate locations. This paper deals with methods focused on Uniform Resource Locator (URL) features for detecting phishing websites. To obtain a deeper understanding of phishing URLs structure, we consider different data mining algorithms for evaluating the features.*

Key Words: Phishing, benign, URL, Machine Learning, Deep Learning

1. INTRODUCTION

Phishing involves the use of both social engineering and technologically advanced skills to obtain and misuse sensitive and personal data from customers and their financial account credentials. Rather than any other segment of the industry, phishing targets the digital market. By either copying or changing the legitimate page a little bit, phishing is carried out by creating a new website such that the online user cannot distinguish between the fake and legitimate pages. There are various kinds of phishing attack domains, such as online payments, websites, email and financial institutions, file hosting or cloud storage, and many more, can occur.

Much of the purchases are actually made electronically by individuals. All is done through websites to pay the bills or pass money. Once these stolen credentials are obtained by phishers, they may use this information to build a false victim account that has a significant effect on their credentials or can refuse users access to their accounts.

1.1 Phishing Technique

First of all, criminals who want to get and misuse confidential data design fake to manipulated as same of a legit web page and electronic mail, basically from a money related organization. Using brand elements of a legitimate corporation, the e-mail will then be designed. In the recent years Internet has developed and grown rapidly as a medium of communication is the ease of website creation, which also allows the misuse of corporate brand elements which users do seek for authentication mechanisms. In an attempt to lure them into the scheme, Phisher then forwards the manipulated e-mails to many individuals. Innocent users are then carried to a web page, when they click the link within one of these manipulated e-mails which looks shockingly similar to the genuine website.

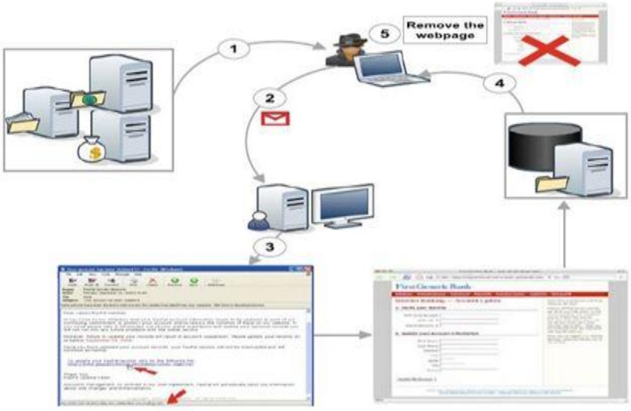


Fig -1: Phishing life cycle 1) phisher copies the content from legitimate site and constructs the phishing site; 2) phisher sent link of phishing URL to Internet user; 3) user opens the link and fills personal credentials on fake site; 4) phisher steals the personal information of user; 5) phisher deleted the fake web page; [1]

To set up phishing web sites, attackers routinely use a collection of software tools known as phishing kits. Phishing can be deployed by people with very little technical expertise through phishing kits. A phishing package includes a component for the website and a component for data processing. In order to create a phishing website, the website portion consists of images, codes and various content materials. The statistics obtained from the data processing tool (password, login time, IP) are registered and sent to the attacker. Phishing kits are aimed predominantly at banking, financial institutions, retail and consumer goods companies such as Microsoft, PayPal, Amazon, Apple, etc.

1.2 Specific Anti-Phishing Technologies

Different approaches and unique technologies are being developed to fight phishing. Using a single technology, phishing cannot be terminated. The correct implementation and changes in the current security technologies, however, will significantly reduce the occurrence of phishing and also the losses suffered from it.

In recent decades, the want of successful action taken for counter has made phishing identification a mass discussion of study. As a outcome, 3 crucial kind of phishing identification proposals have emerged: (a) Techniques based on boycott and believed trustworthy [2], [3] (b) Approaches based on visual similarities of web pages [4] (c) Approaches based on features of URL and website content [5].

In identifying recently phishing webpages that the software doesn't seem to be changed, thus blacklist method is unsuccessful. The technique focused on visible likeness takes out visible features from phishing websites and further applies these characteristics to classify web pages for phishing. Therefore, any web page information distortion affects the retrieval of visual content, leading to misclassification. The URL and web content functionality are used by most existing phishing detection approaches to differentiate between phishing and legitimate websites, e.g. [5], [6]. To enhance detection efficiency and allow zero-day phishing protection, algorithm of machine learning was also been clubbed with URL and web information trait.

1.2 Statistics of Phishing attacks

Phishing has grown successively from recent few years becoming top most types of internet identity and financial theft scams that cause personal, social and economic harm. In the first quarter of 2019, phishing accounted for 29 per cent of all fraud attacks and India was second in the list of top phishing hosting countries to the US.

The total number of identified phishing sites in 2019 is 146,994, according to the Anti-Phishing Working Group (APWG Q2 2020) survey [7]. Webmail and Software-as-a-Service (SaaS) are the most targeted business sectors here. Financial services, transportation, cloud storage services, SAAS/Webmail and payment services are the subject of 80 percent of attacks. The most attractive goal for phishing is the payment market [7].

Table -1: Statistical Highlights for Q2 2020 [7]

	April	May	June
Number of unique phishing Websites and web pages detected	48,951	52,007	46,036
Number of unique phishing e-mail reports	43,282	39,308	44,497
Number of brands which were targeted by phishing campaigns	364	352	363

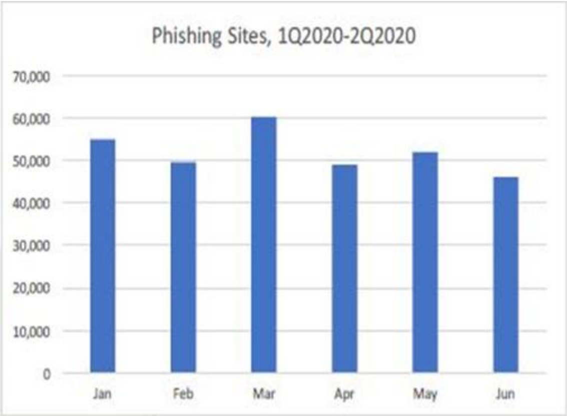


Chart -1: Phishing Sites Trend [7]

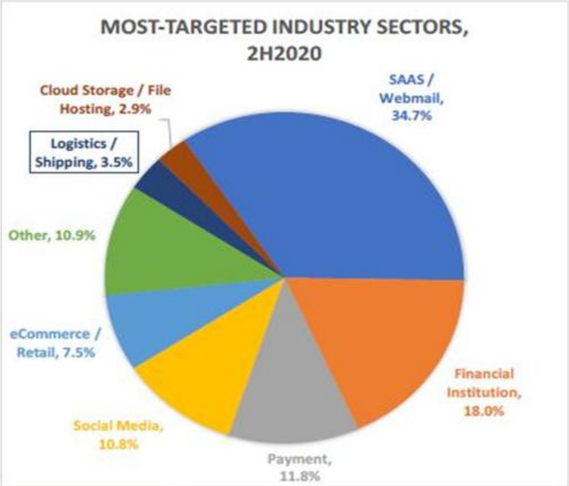


Chart -2: Most Targeted Industry Sectors [7]

2. RELATED WORK

The statistics of suspicious URLs have been analyzed in some way by many researchers. Our strategy takes significant plan over already done research. We study the already done research in the phishing page identification using URL trait that guided our self-developed technique.

In order to identify phishing URLs, work by Garera et al. [8] that tries logistic regression on manually selected traits. The traits have the incorporation of keywords in the URL with red colour flags, traits focused on Google's Page Rank, and recommendations for web page quality from Google.

McGrath and Gupta [9] did not create a classifier, but conducted a comparative study of data sets related to phished and un-phished URLs. They collate un-phished URLs obtained from the DMOZ Open Directory Project with PhishTank phishing URLs. IP addresses, WHOIS thin records containing date and registrar-provided data, geographic information, and lexical URL features such as length, distribution of characters, and existence of predefined brand names are the features they examine.

By classifying them with URL attributes such as length, number of special characters, directory, domain name, and file name, Le, Markopoulou, and Faloutsos [10] detected phishing websites. Using Support Vector Machines, the framework classifies websites offline. For online classification, adaptive Regularization of Weights, Confidence Weighted, and Online Perceptron are used. Using the Adaptive Regularization of Weights algorithm improves the accuracy rate according to the results of the experiments, thus reducing the need for machine resources.

Ma et al. [11], [12] compared many batch-related learning algorithms to identify phishing URLs and demonstrated that the highest classification accuracy comes from the combination of lexical traits and host-related. They also collate the accuracy of batch-related technique to internet technique when utilizing all traits and discovered that batch-situated algorithms do outperform online algorithms, especially Confidence-Weighted (CW).

3. PROBLEM OVERVIEW

Often referred to as web links, URLs are the primary means by which users find information on the Internet. By analysing the lexical features of URLs, we aim to derive classification and predictive models that detect and classify phishing websites.

4. BACKGROUND

The proposed framework is being trained and tested using some classifiers and neural networks. For processing the feature set, the machine learning algorithms considered are:

4.1 Decision Tree Algorithm:

A predictive machine-learning model that determines a new sample's target value (dependent variable) based on the various attribute values of the data available. Using tree representation, the decision tree algorithm attempts to resolve the problem. An attribute corresponds to each internal node of the tree, and a class label corresponds to each leaf node. Such as Microsoft, PayPal, Amazon, Apple, etc.

4.2 Random Forest Algorithm:

As its name suggests, Random Forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree spits out a class prediction in the random forest and the class with the most votes becomes the prediction of our model. The basic idea behind random forests is the wisdom of crowds, a plain but strong one. So, the prerequisites for a well-performing random forest are:

1. In our features, there needs to be some real signal so that models constructed using those features perform better than random guessing.

2. The predictions made by the individual trees (and thus the errors) need to have low correlations with each other.

4.3 Support Vector Machine Algorithm:

One of the most common classifications these days is Support Vector Machines (SVM). The concept here is to set the ideal separating hyperplane between two classes by maximising the margin between the closest points of the classes.

Although SVMs are very powerful and widely used in classification, there are many disadvantages to them. In order to train the data, they require high computations. They are also vulnerable to noisy data and therefore susceptible to overfitting.

4.4 Bagging Classifier:

Bagging is used where the purpose is to reduce a decision tree classifier's variance. The goal here is to produce many subsets of data with substitution from the training sample picked randomly.

For the preparation of the decision trees, each array of subset data is used. We get an ensemble of various models as a result. The average, which is more stable than a single decision tree classifier, is used for all the projections from various trees.

4.5 AdaBoosting Classifier:

AdaBoost, short for Adaptive Boosting, is an AI meta-calculation figured by Yoav Freund and Robert Schapire. It tends to be utilized related to numerous different kinds of learning calculations to improve execution. The yield of the other learning calculations ('frail students') is joined into a weighted aggregate that speaks to the last yield of the supported classifier.

AdaBoost is versatile as in resulting feeble students are changed for those examples misclassified by past classifiers. AdaBoost is delicate to boisterous information and anomalies.

4.6 Logistic Regression:

For binary data (0/1 response) prediction, logistic regression is the most commonly used statistical model in many fields. Due to its simplicity and great interpretability, it has been widely applied. It usually uses the logit function as a part of generalised linear models. When the relationship in the data is roughly linear, logistic regression performs well. However, if complicated nonlinear interactions occur between the variables, it performs poorly. Furthermore, it needs more statistical assumptions than other strategies before being implemented. Even, if there is missing information in the dataset, the prediction rate is affected.

4.7 K-Nearest Neighbour (KNN):

K Nearest Neighbour (KNN) is an extremely basic, straightforward, adaptable and one of the highest AI calculations. KNN utilized in the assortment of utilizations, for example, money, medical care, political theory, penmanship discovery, picture acknowledgment and video acknowledgment. KNN calculation utilized for both order and relapse issues. KNN calculation dependent on highlight closeness approach.

K indicates the number of nearest neighbours. Deciding the number of neighbours in a model is the crucial factor which need to be looked upon. K is generally an odd number if the number of classes is 2. When K=1, then the algorithm is known as the nearest neighbour algorithm.

4.8 Pipeline:

In a Machine Learning (ML) model, there are several moving parts that have to be put together for an ML model to effectively perform and generate results. This method of connecting various sections of the ML mechanism together is known as a pipeline. A pipeline for a Data Scientist is a simplified but quite significant term.

4.9 Naïve-Bayes Algorithm:

The probabilistic classifier, based on Bayes ' theorem with the assumption of "naive" independence, is Naïve Bayes. Used in text categorization, this classifier can be an earning-based version of keyword filtering. They are among the least difficult Bayesian organization models. But they could be combined with Kernel thickness assessment and accomplish higher exactness levels.

4.10 CNN (Convolutional Neural Networks):

CNN be in the ANN class of computational models influenced by biological neural network characteristics. A Convolutional NN is a deep learning algorithm that executes fine to classify basic swatch in the information that can then be used in subsequent layers to construct more complex patterns. For the construction of CNNs, 2 kinds of layers are normally utilized; pooling layers and convolutional layers. The, pooling layer's role is to collate exact features semantically into 1 while convolutional layer's works is to identify local club of features from the last layer [14].

In spite of the fact CNN is usually often used in a multidimensional fashion and has thus found favorable outcome in photos and visual analysis-related situations, they can also be used to 1-D data. Datasets that hold a 1-D structure can be functioned using a 1-D (CNN). Basic dispute among a 1-Dimension and a 2-Dimension or 3-Dimension CNN is the input information structure and how the filters (feature detector) slides through the dataset. For 1D CNN, the filters only slide across the input data in one direction.

4.11 Artificial Neural Network (ANN)

A neural network is structured as a collection of identical units (neurons) that are interconnected. The interconnections are used from one neuron to the other to transmit signals. In addition, to boost the delivery between neurons, the interconnections have weights. Neurons are not powerful themselves, but they can perform complicated computations when linked to others. When the network is educated, weights on the interconnections are changed, so significant interconnections play an increased role during the testing process. The neural network in the figure consists of one layer of input, one hidden layer, and one layer of output. The network is called feedforward, because interconnections do not loop back or bypass other neurons. The strength of neural networks comes from the hidden neurons' nonlinearity. Consequently, in order to be able to learn complex mappings, it is necessary to incorporate nonlinearity into the network. The commonly used function in neural network research is the sigmoid function, which has the form:

$$a(x) = \frac{1}{1+e^{-x}}$$

Although competitive in learning ability, the fitting of neural network models requires some experience, since multiple local minima are standard and delicate regularization is required.

5. DESIGN FLOW

The work consists of extracting collected URLs and interpretation based on the page and lexical function. The first stage is the collection of harmless URLs and phishing. In order to shape a database of feature values, popularity-based and lexical-based feature extractions are used. Information mined using different methods of machine learning is the database. A unique classifier and neural network are selected after evaluation of the classifiers and neural networks and is implemented.

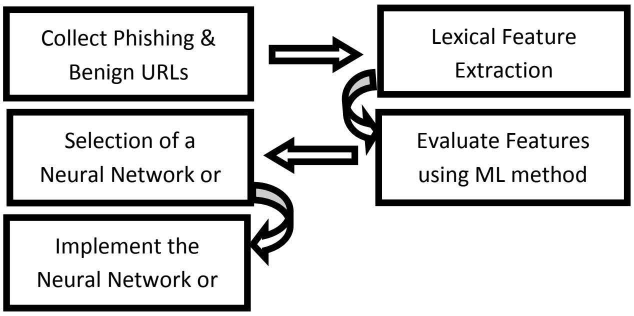

Collect Phishing & Benign URLs
Lexical Feature Extraction
Evaluate Features using ML method
Selection of a Neural Network or
Implement the Neural Network or

Fig -2: Design Flow Graph

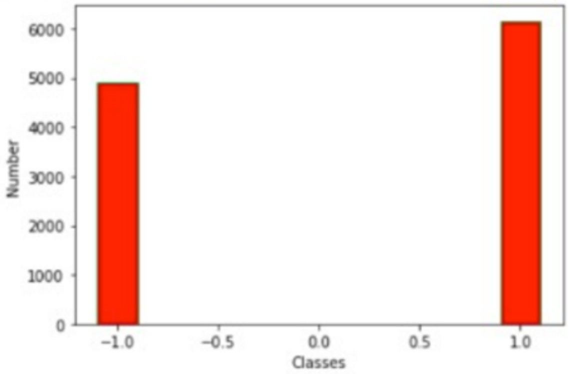


Chart -3: Classes Distribution

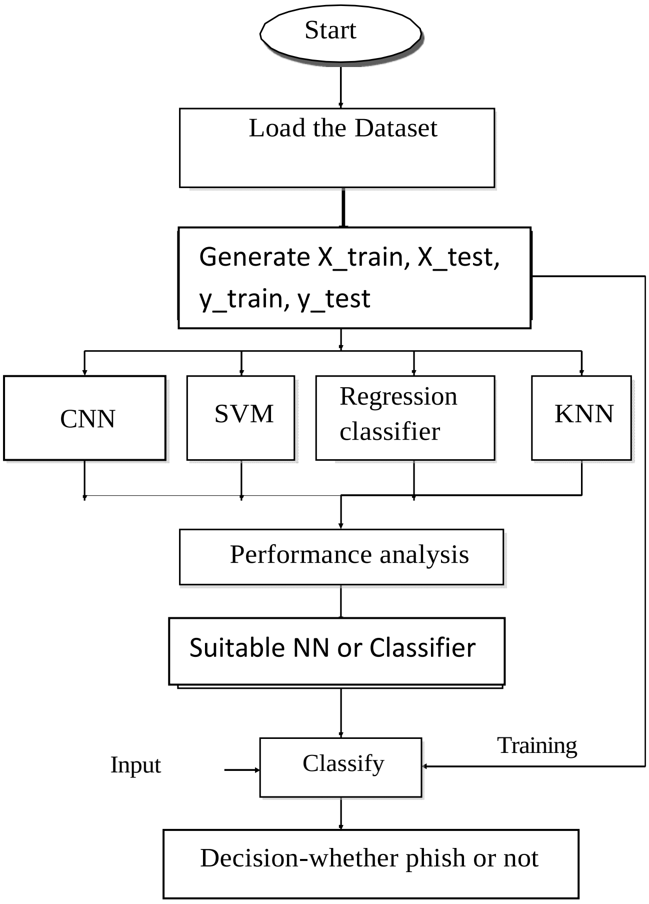

Start
Load the Dataset
Generate X_train, X_test, y_train, y_test
CNN
SVM
Regression classifier
KNN
Performance analysis
Suitable NN or Classifier
Input
Classify
Training
Decision-whether phish or not

Fig -3: Program Flow

5.1 Dataset Used:

The dataset for phishing URLs is downloaded from UCI ML Repository [13]. The dataset consists of 11055 samples/rows and 31 columns. The data is divided into two classes [-1, 1]. The dataset consists of 4898 Phished [-1] samples and 6157 Non- Phished [1] samples. Further, the dataset consists of 30 features which are show in the Table 2 below.

Table -2: Dataset Feature Summary

Attribute No.	Attributes	Possible vals
1	Having_IP_Address	-1,1
2	URL_Length	1,0,-1
3	Shortening_Service	1,-1
4	having_At_Symbol	1,-1
5	double_slash_redirecting	-1,1
6	Prefix_Suffix	-1,1
7	having_Sub_Domain	-1,0,1
8	SSLfinal_State	-1,1,0
9	Domain_registration_length	-1,1
10	Favicon	1,-1
11	Port	1,-1
12	HTTPS_token	-1,1
13	Request_URL	-1,1
14	URL_of_Anchor	-1,0,1
15	Links_in_tags	1,-1,0
16	SFH (server form handler)	-1,1,0
17	Submitting_to_email	-1,1
18	Abnormal_URL	-1,1
19	Redirect_page	0,1
20	onMouseOver (using to hide link)	1,-1
21	RightClick	1,-1
22	Using pop-up window	1,-1
23	Iframe	1,-1
24	age_of_domain	-1,1
25	DNSRecord	-1,1
26	web_traffic	-1,0,1
27	Page_Rank	-1,1
28	Google_Index	-1,1
29	Links_pointing_to_page	1,0, -1
30	Statistical_report	-1,1
Class	Result	-1,1

5.2 Page / Popularity Based Property

Features of popularity indicate how popular a web page is among users of the Internet. The following are different popularity characteristics:

a) PageRank: It is one of the methods used by Google to determine the relevance or importance of a page. When Google does its re-indexing, the maximum PR of all web pages changes each month.

The Page Ranks form a distribution of probability over web pages, so the sum of Page Ranks for all web pages is equal to 1.

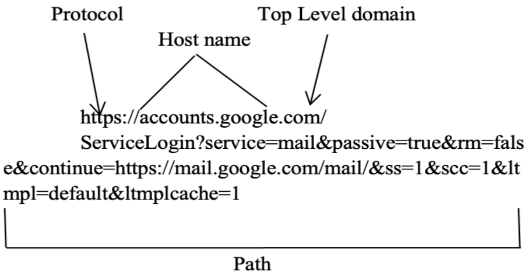
b) Traffic Details: A site's popularity is determined by amount of traffic they get. Traffic close to 1 is precise. Ranks over 100,000 are not so reliable as there is a high risk of error.

5.3 Analysis of Lexical Features

Lexical traits are the text related identities of the URL itself. URLs are text strings that can be understood by humans. Browsers are made specifically to translate URLs into instructions which further locate the server hosting the site and loads the material which is in within these websites. URLs have the following standard syntax to facilitate this machine translation process.

<protocol>://<hostname><path>

Particular example of URL resolution is shown below:



https://accounts.google.com/ServiceLogin?service=mail&passive=true&rm=false&continue=https://mail.google.com/mail/&ss=1&scc=1&ltmpl=default&ltmplcache=1

The < protocol > portion of the URL gives us the idea of which network protocol to retrieve the requested resource should be used. HTTP, HTTPS, FTP are the most frequently used protocols. < hostname > is a server identifier for server delivering the web content which is their on the Internet. Internet Protocol (IP) address, often represented as a human-readable domain name written in normal English, particularly for the user's understanding. The URL's < path > is path directory referring to a file which is located on a local machine. The route tokens separated by the various punctuation marks such as "/", ".", and "-", indicates mostly how the web structuring has taken place.

The outcomes of these trials are talked about in the upcoming segment. In sequence to calculate model accuracy, we used the following metrics: Precision, accuracy, F1-score and recall. The metrics are discussed as below:

- Precision: All true positives divided by all positive predictions. i.e. Was the model right when it predicted positive? Given by:
  Precision = TP/(TP+FP)
- Accuracy: Defined as the ratio between correctly predicted outcomes and the sum of all predictions. It is given by:
  Accuracy = (TP+TN)/(TP+FP+FN+TN)
- F-1 Score: This is the weighted average of precision and recall, given by:
  F1-score = (2*(Precision*Recall))/((Precision+Recall))
- Recall: True positives divided by all actual positives i.e. how many positives did the model identify out of all possible positives? Given by:
  Recall = TP/(TP+FN)

6. RESULT AND DISCUSSIONS

Our preliminary work's key results include:

- There is a significant amount of feature distinction that Phishing URLs and domains show from other domains and URLs.
- There is a difference in lengths of genuine URLs and domain names on the Internet, to phishing URLs and domain names.
- The name of the brand they targeted included several of the phishing URLs.

The URL feature dataset has been analysed using Naïve Bayes, k-NN, SVM, Decision Tree, Logistic Regression classifying algorithms. In addition to this we also implemented perceptron, pipeline neural network, ANN and CNN over the dataset and also trained the dataset with the ensemble learning with Adaboost and Bagging methods. We have kept the portion snap for our models to 30% i.e., 70% portion of the information is taken as training set and 30% portion as test set. Evaluation of performance is based on Accuracy Score, Recall Score, F1 score and Precision Score. The result is tabulated in TABLE III. We can see that among all the algorithms we have used the best accuracy is given by the Bagging algorithm which gives an accuracy of 97.015. payment market [7].

Table -3: Metric Scores of Different Algorithms

	Accuracy Score	F1-Score	Precision Score	Recall Score
Adaboost	96.623	97.029	96.721	97.339
Bagging	97.015	97.346	97.06	97.634
DT	95.628	96.119	95.582	96.663
RF	96.834	97.149	96.234	98.081
LR	92.583	93.512	92.199	94.863
KNN	94.271	94.873	94.566	95.181
NB	59.632	44.045	100	28.242
SVM	93.367	94.092	93.444	94.753
Perceptron	90.925	92.001	88.906	95.319
Pipeline	87.277	88.507	90.832	86.298
CNN	92.612	90.611	90.455	91.223
ANN	33.076	32.213	30.455	29.01

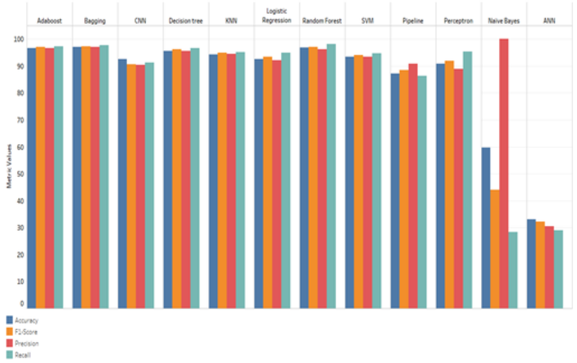


Chart -4: Comparison of different algorithms

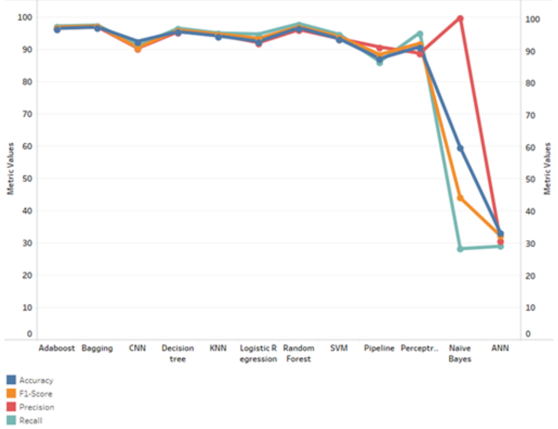


Chart -5: Using Line graph to see relative performance

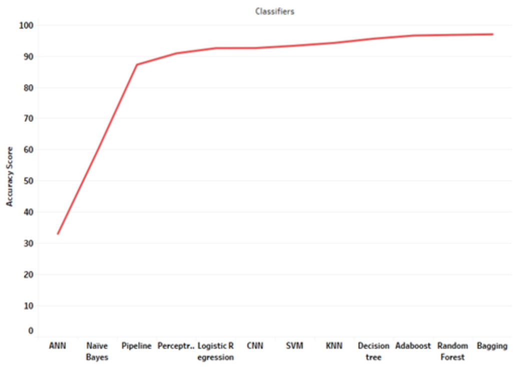


Chart -6: Accuracy Scores of Different Algorithms

7. CONCLUSION AND FUTURE WORK

Our suggested phishing detection system focused on the detection of phishing URLs using a machine learning approach was discussed in this paper. By using different machine learning algorithms, such as Decision Tree, KNN, Random Forest, and Naive Bayes, along with some deep learning algorithms like CNN, ANN, etc (for comparison) we have implemented a phishing detection system. Using different data mining algorithms, multiple features are compared. The results points to the efficiency that the lexical features can achieve when used.

In this area, a specific challenge is that criminals are frequently making new tactics to combat our security steps. To succeed, we need algorithms that continually are adapting to new examples and features of phishing URLs. We look forward to improve the model training process by hyper tuning the features and parameters and also in analysing the various aspects of online learning and collecting the different data to understand the new trends in the phishing activities. Further, for our deep learning models we aim to improve the model training process by implementing the automating selection of significant parameters which will result in optimal performance.

REFERENCES

[1] Ankit Kumar Jain and B. B. Gupta, "PHISH-SAFE: URL Features-Based Phishing Detection System Using Machine Learning"

[2] A. K. Jain and B. B. Gupta, "A novel approach to protect against phishing attacks at client side using auto-updated white-list,"

[3] P. Prakash, R. Rao Kompella, M. Gupta, M. Kumar, "Phishnet: predictive blacklisting to setect phishing attacks,"

[4] A. K. Jain and B. B. Gupta, 'Phishing Detection: Analysis of Visual Similarity Based Approaches', Security and Communication Networks

[5] R. M. Mohammad, L. McCluskey, and F. Thabtah, "Intelligent rulebased phishing websites classification," IET Information Security

[6] S. C. Jeeva and E. B. Rajsingh, "Intelligent phishing URL detection using association rule mining"

[7] APWG Q2 2019 REPORT, https://docs.apwg.org/reports/apwg_trends_report_q2_2020.pdf

[8] Garera S., Provos N., Chew M., Rubin A. D., "A Framework for Detection and measurement of phishing attacks"

[9] M. Gupta, D. K. McGrath, "Behind Phishing: An Examination of Phisher Modi Operandi"

[10] Le, A., Markopoulou, A., & Faloutsos, M. (2011). Phishdef: URL names say it all. In 2011 Proceedings IEEE INFOCOM, 2011 (pp. 191–195)

[11] J. Ma, L. K. Saul, S. Savage, G. M. Voelker," Beyond Blacklists: Learning to Detect Phishing Web Sites from Suspicious URLs"

[12] J. Ma, L. K. Saul, S. Savage, G. M. Voelker," Learning to Detect Phishing URLs", ACM Transactions on Intelligent Systems and Technology

[13] UCI Machine Learning Repository, "Phishing Dataset" https://archive.ics.uci.edu/ml/datasets/phishing+websites

[14] Y. LeCun, Y. Bengio, and G. Hinton, Deep learning, Nature 521 (2015), no. 7553, 436-444