IJNRD.ORG ISSN : 2456-4184
INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD) | IJNRD.ORG
An International Open Access, Peer-reviewed, Refereed Journal

# Detection of Phishing Websites Using Machine Learning

**[1]Dr.M.Hemalatha, [2]A. Mekala**

Associate Professor, PG & Research Department of Computer Science
Sri Ramakrishna College Arts &Science(Autonomous), Coimbatore, India

UG Student, PG & Research Department of Computer Science
Sri Ramakrishna College Arts &Science( Autonomous), Coimbatore, India

***Abstract***: Phishing remains a unavoidable danger, abusing human helplessness to trap people into disclosing delicate data through misleading websites. To combat this, leveraging anti-phishing methods is vital, with machine learning rising as a strong apparatus. In a later approach, the Slope Boosting Classifier show has been utilized to recognize between true blue and phishing URLs based on different critical highlights. By analyzing and comparing key properties, this strategy successfully recognizes false websites in real-time, advertising a promising defense against advancing phishing strategies encouraged by progressing innovation.

## I.INTRODUCTION

Consumers have lost billions of dollars each year as a result of phishing operations. Refers to thieves' tricks for obtaining private information from a group of unwitting Internet users. Fraudsters obtain personal and financial account information such as usernames and passwords using fake email and phishing software to steal sensitive information.

This research examines strategies for detecting phishing Web sites using machine learning techniques to analyse various aspects of benign and phishing URLs. It investigates how linguistic cues, host features, and page significance attributes are used to identify phishing site. The fine-tuned parameters aid in the selection of the most appropriate machine learning method for distinguishing between phishing and benign sites. Criminals that seek to steal sensitive information first establish illegal duplicates of legitimate websites and e-mail accounts, frequently from financial institutions or other companies that deal with financial data. The e-mail will be made up of real firm logos and slogans.

One of the reasons for the rapid growth of the internet as a means of communication is that it allows the misuse of trademarks, brand names, and other corporate identities that consumers rely on as verification processes. "Spoof" e-mails are sent to many people in order make them involved in the criminal deception. Consumers are paid on a fraudulent website that appears to come from the real company when these emails are opened or when a link is clicked on the email.

## II.DATA COLLECTION

In the first module we develop the data collection process. This is the first real step towards the real development of a machine learning model, collecting data. This is a critical step that will cascade in how good the model will be, the more and better data that we get; the better our model will perform.

There are several techniques to collect the data, like web scraping, manual interventions. The dataset is referred from the popular dataset repository called kaggle. The following is the dataset link for the Detection of Phishing Websites Using Machine Learning.

**Kaggle Dataset Link**:

https://www.kaggle.com/datasets/jayaprakashpondy/phishing-websites-feature-dataset

| **Index** | **index id** |
|---|---|
| **UsingIP** | (categorical - signed numeric) : { -1,1 } |
| **LongURL** | (categorical - signed numeric) : { 1,0,-1 } |
| **ShortURLC** | (categorical - signed numeric) : { -1,1 } |
| **Symbol@** | (categorical - signed numeric) : { -1,1 } |
| **Redirecting** | (categorical - signed numeric) : { -1,1 } |
| **PrefixSuffix** | (categorical - signed numeric) : { -1,1 } |
| **SubDomains** | (categorical - signed numeric) : { -1,0,1 } |
| **HTTPS** | (categorical - signed numeric) : { -1,1,0 } |
| **DomainRegLen** | (categorical-signed numeric) : { 1,-1 } |
| **Favicon** | (categorical-signed numeric) : { 1,-1 } |
| **NonStdPort** | (categorical-signed numeric) : { 1,-1 } |
| **HTTPSDomainURL** | (categorical-signed numeric) : { 1,-1 } |
| **RequestURL** | (categorical-signed numeric) : { 1,-1 } |
| **AnchorURL** | (categorical - signed numeric) : { -1,0,1 } |
| **LinksInScriptTags** | (categorical - signed numeric) : { -1,0,1 } |
| **ServerFormHandler** | (categorical - signed numeric) : { -1,0,1 } |
| **InfoEmail** | (categorical - signed numeric) : { -1,1 } |
| **AbnormalURL** | (categorical - signed numeric) : { -1,1 } |
| **WebsiteForwarding** | (categorical - signed numeric) : { 0,1 } |
| **StatusBarCust** | (categorical - signed numeric) : { -1,1 } |
| **DisableRightClick** | (categorical - signed numeric) : { -1,1 } |
| **UsingPopupWindow** | (categorical - signed numeric) : { -1,1 } |
| **IframeRedirection** | (categorical - signed numeric) : { -1,1 } |
| **AgeofDomain** | (categorical - signed numeric) : { -1,1 } |
| **DNSRecording** | (categorical - signed numeric) : { -1,1 } |
| **WebsiteTraffic** | (categorical - signed numeric) : { -1,0,1 } |
| **PageRank** | (categorical - signed numeric) : { -1,1 } |
| **GoogleIndex** | (categorical - signed numeric) : { -1,1 } |
| **LinksPointingToPage** | (categorical - signed numeric) : { -1,0,1 } |
| **StatsReport** | (categorical - signed numeric) : { -1,1 } |
| **Class** | (categorical - signed numeric) : { -1,1 } |

### II. I DATA PREPARATION:

Wrangle data and prepare it for training. Clean that which may require it (remove duplicates, correct errors, deal with missing values, normalization, and data type conversions, etc.).

Randomize data, which erases the effects of the particular order in which we collected and/or otherwise prepared our data.

Visualize data to help detect relevant relationships between variables or class imbalances (bias alert!), or perform other exploratory analysis Split into training and evaluation sets.

### II. II MODEL SELECTION:

We used Gradient Boosting Classifier machine learning algorithm. We got an accuracy of training Accuracy 98.9% so we implemented this algorithm.

### III.Gradient Boosting Classifier Algorithm:

**Step -1**

The first step in gradient boosting is to build a base model to predict the observations in the training dataset. For simplicity we take an average of the target column and assume that to be the predicted value as shown below:

Why did I say we take the average of the target column? Well, there is math involved behind this. Mathematically the first step can be written as:

$$F_0(x) = \arg\min_{\gamma} \sum_{i=1}^{n} L(y_i, \gamma).$$

Looking at this may give you a headache, but don’t worry we will try to understand what is written here.

Here L is our loss function Gamma is our predicted value Arg min means we have to find a predicted value/gamma for which the loss function is minimum.

Since the target column is continuous our loss function will be:

$$L = \frac{1}{n}\sum_{i=0}^{n}\left(y_i - \gamma_i\right)^2$$

Here is the observed value And gamma is the predicted value

Now we need to find a minimum value of gamma such that this loss function is minimum. We all have studied how to find minima and maxima in our 12th grade. Did we use to differentiate this loss function and then put it equal to 0 right? Yes, we will do the same here.

$$\frac{\mathrm{dL}}{\mathrm{d}\gamma} = \frac{2}{2}\left(\sum_{i=0}^{n}\left(y_i - \gamma_i\right)\right) = -\sum_{i=0}^{n}\left(y_i - \gamma_i\right)$$

Let's see how to do this with the help of our example. Remember that y_i is our observed value and gamma is our predicted value, by plugging the values in the above formula we get:

Hence for gamma=14500, the loss function will be minimum so this value will become our prediction for the base model.

**Step-2**

The next step is to calculate the pseudo residuals which are (observed value – predicted value) again the question comes why only observed – predicted? Everything is mathematically proved, let's from where did this formula come from. This step can be written as:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1, \ldots, n.$$

Here F(xi) is the previous model and m is the number of DT made. We are just taking the derivative of loss function w.r.t the predicted value and we have already calculated this derivative:

$$\frac{dL}{d\gamma} = -\left(y_i - \gamma_i\right) = -(Observed - Predicted)$$

If you see the formula of residuals above, we see that the derivative of the loss function is multiplied by a negative sign, so now we get:

$$(Observed - Predicted)$$

The predicted value here is the prediction made by the previous model. In our example the prediction made by the previous model (initial base model prediction) is 14500, to calculate the residuals our formula becomes:

$$(Observed - 14500)$$

In the next step, we will build a model on these pseudo residuals and make predictions. Why do we do this? Because we want to minimize these residuals and minimizing the residuals will eventually improve our model accuracy and prediction power. So, using the Residual as target and the original feature Cylinder number, cylinder height, and Engine location we will generate new predictions. Note that the predictions, in this case, will be the error values, not the predicted car price values since our target column is an error now.

let's say hm(x) is our DT made on these residuals.

**Step- 3**

In this step we find the output values for each leaf of our decision tree. That means there might be a case where 1 leaf gets more than 1 residual, hence we need to find the final output of all the leaves. TO find the output we can simply take the average of all the numbers in a leaf, doesn't matter if there is only 1 number or more than 1.Let's see why

do we take the average of all the numbers. Mathematically this step can be represented as:

$$\gamma_m = \arg\min_{\gamma} \sum_{i=1}^{n} L\left(y_i, F_{m-1}(x_i) + \gamma h_m(x_i)\right).$$

Here hm(xi) is the DT made on residuals and m is the number of DT. When m=1 we are talking about the 1st DT and when it is "M" we are talking about the last DT. The output value for the leaf is the value of gamma that minimizes the Loss function. The left-hand side "Gamma" is the output value of a particular leaf. On the right-hand side [Fm-1(xi)+ɣhm(xi))] is similar to step 1 but here the difference is that we are taking previous predictions whereas earlier there was no previous prediction. Let's understand this even better with the help of an example. Suppose this is our regressor tree:

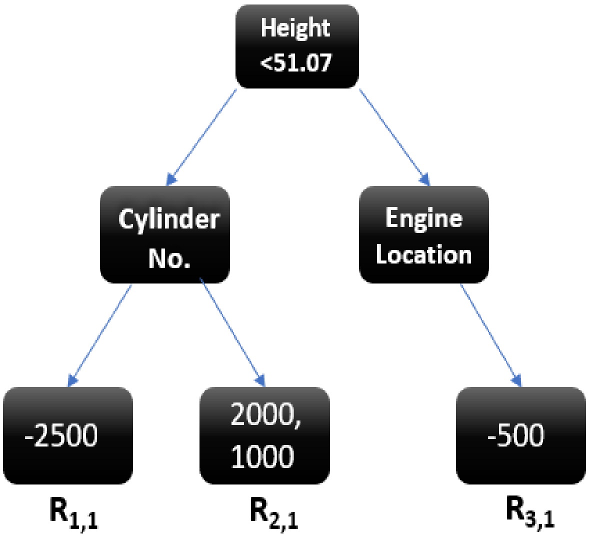


**Image 1**

We see 1st residual goes in R1,1 ,2nd and 3rd residuals go in R2,1 and 4th residual goes in R3,1 . Let's calculate the output for the first leave that is R1, 1

$$\gamma_{1,1} = argmin\frac{1}{2}(12000 - (14500 + \gamma))^2$$

$$\gamma_{1,1} = argmin\frac{1}{2}(-2500 - \gamma)^2$$

Now we need to find the value for gamma for which this function is minimum. So we find the derivative of this equation w.r.t gamma and put it equal to 0.

$$\frac{d}{d\gamma}\frac{1}{2}\left(-2500-\gamma\right)^2 = 0$$

$$-2500 - \gamma = 0$$

$$\gamma = -2500$$

Hence the leaf R1,1 has an output value of -2500. Now let's solve for the R2,1 Let's take the derivative to get the minimum value of gamma for which this function is:

$$\frac{d}{d\gamma}\left[\frac{1}{2}(2000-\gamma)^2+\frac{1}{2}\left(1000-\gamma\right)^2\right] = 0$$

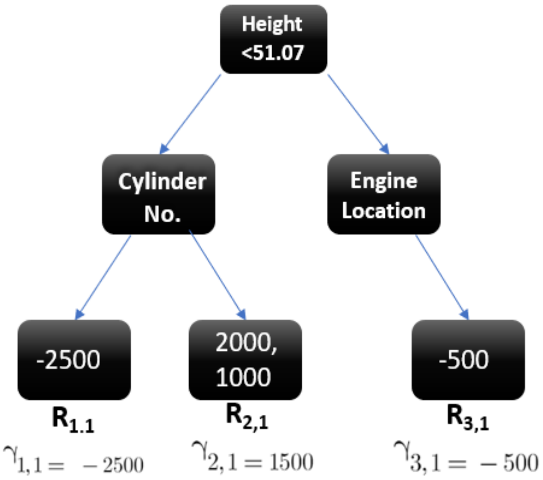
$$2000-\gamma+1000-\gamma=0$$

$$3000-2\gamma=0$$

$$\frac{3000}{2}=\gamma$$

$$\gamma=1500$$

We end up with the average of the residuals in the leaf R2,1 . Hence if we get any leaf with more than 1 residual, we can simply find the average of that leaf and that will be our final output. Now after calculating the output of all the leaves, we get:



**Step-4**

This is finally the last step where we have to update the predictions of the previous model. It can be updated as:

Update the model:

$$F_m(x) = F_{m-1}(x)+\nu_m h_m(x)$$

Where m is the number of decision trees made. Since we have just started building our model so our m=1. Now to make a new DT our new predictions will be:Here Fm-1(x) is the prediction of the base model (previous prediction) since F1-1=0 , F0 is our base model hence the previous prediction is 14500.

nu is the learning rate that is usually selected between 0-1. It reduces the effect each tree has on the final prediction, and this improves accuracy in the long run. Let’s take nu=0.1 in this example. Hm(x) is the recent DT made on the residuals.

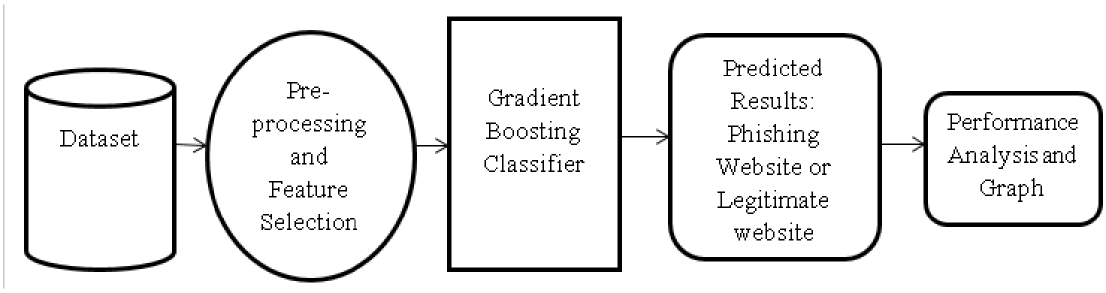
Let’s calculate the new prediction now:

Suppose we want to find a prediction of our first data point which has a car height of 48.8. This data point will go through this decision tree and the output it gets will be multiplied with the learning rate and then added to the previous prediction.

Now let’s say m=2 which means we have built 2 decision trees and now we want to have new predictions. This time we will add the previous prediction that is F1(x) to the new DT made on residuals. We will iterate through these steps again and again till the loss is negligible. I am taking a hypothetical example here just to make you understand how this predicts for a new dataset: Image Source : Image 2

If a new data point says height = 1.40 comes, it'll go through all the trees and then will give the prediction. Here we have only 2 trees hence the datapoint will go through these 2 trees and the final output will be F2(x).

## IV .SYSTEM ARCHITECHTURE


Dataset
Pre-processing and Feature Selection
Gradient Boosting Classifier
Predicted Results: Phishing Website or Legitimate website
Performance Analysis and Graph

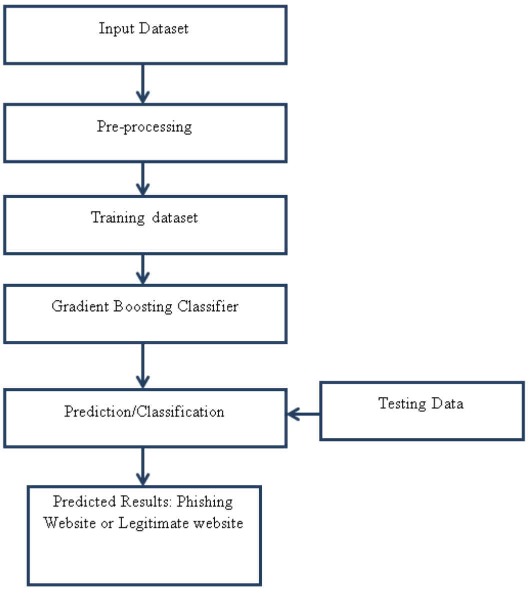
## SYSTEM FLOW DIAGRAM

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.

The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.

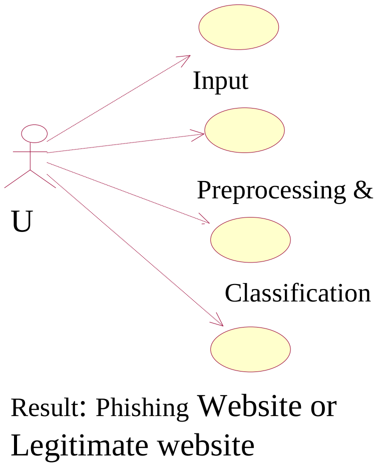
DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.

DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.


Input Dataset
Pre-processing
Training dataset
Gradient Boosting Classifier
Prediction/Classification
Testing Data
Predicted Results: Phishing Website or Legitimate website

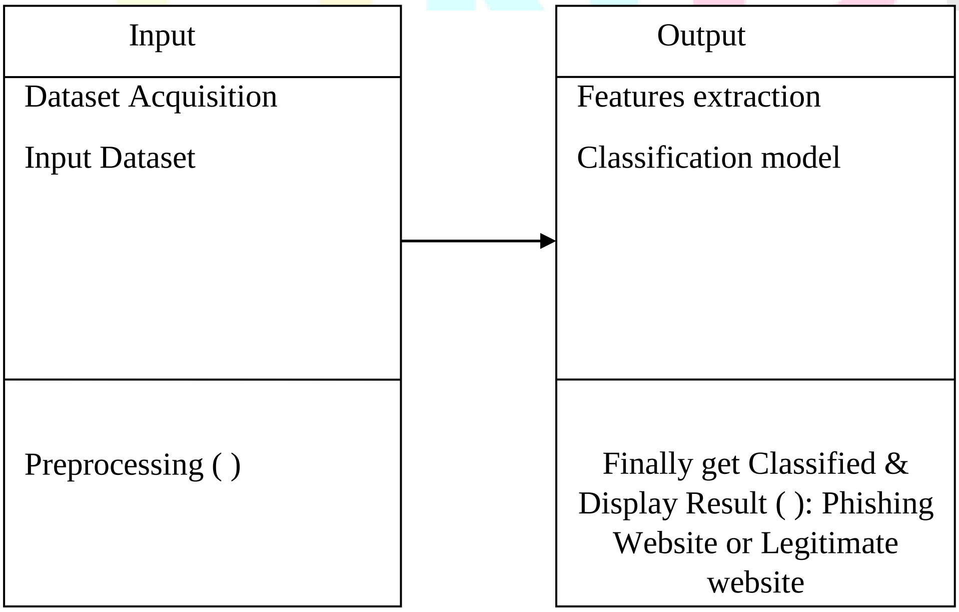
## USE CASE DIAGRAM:

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



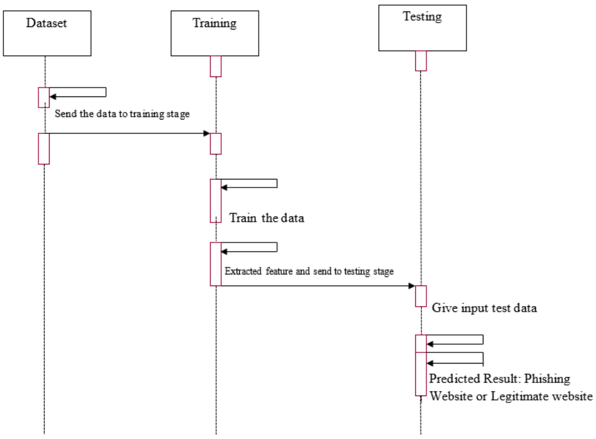
## CLASS DIAGRAM

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



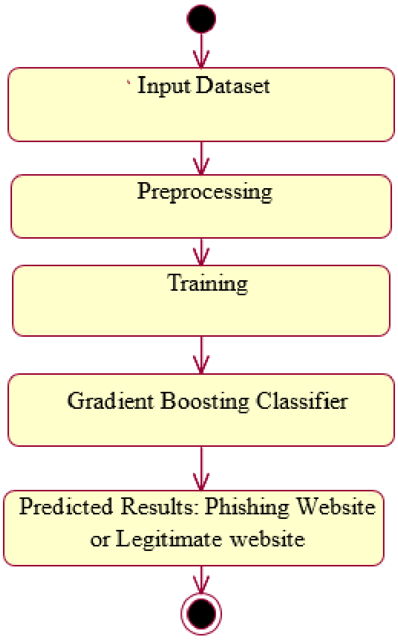
## SEQUENCE DIAGRAM:

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.


Dataset
Training
Testing
Send the data to training stage
Train the data
Extracted feature and send to testing stage
Give input test data
Predicted Result: Phishing Website or Legitimate website

## ACTIVITY DIAGRAM:

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.


Input Dataset
Preprocessing
Training
Gradient Boosting Classifier
Predicted Results: Phishing Website or Legitimate website

## INPUT DESIGN

The input design is the process of entering data to the system. The input design goal is to enter to the computer as accurate as possible. Here inputs are designed effectively so that errors made by the operations are minimized. The inputs to the system have been designed in such a way that manual forms and the inputs are coordinated where the data elements are common to the source document and to the input. The input is acceptable and understandable by the users who are using it.

The quality of the system input determines the quality for system output. Input specification describes the manner in which data entered the system processing.

Input design is the process of converting user-originated inputs to a computer-based format input data are collected and organized into group of similar data. Once identified, appropriate input media are selected for processing

The input design also determines the user to interact efficiently with the system. Input design is a part of overall system design that requires special attention because it is the common source for data processing error. The goal of designing input data is to make entry easy and free from errors.

**Five objectives of the input design are:**

- Effectiveness
- Accuracy
- Ease to use
- Consistency
- Attractiveness

**The main objectives that are done during the input design are:**

- Data is collected from the source
- Transfer of data to an input form is done
- Data is converted to a computer acceptable form
- The converted data is verified
- Data is checked for its accuracy
- Data is transmitted to the computer
- Validation of input data is done
- Data collections are done to eliminate the error

## OUTPUT DESIGN

The output design was done so that results of processing could be communicated to the users. The various outputs have been designed in such a way that they represent the same format that the office and management used to.

Computer output is the most important and direct source of information to the user. Efficient, intelligible output design should improve the systems relationships with the user and help in decision making. A major form of output is the hardcopy from the printer.

Output requirements are designed during system analysis. A good starting point for the output design is the Data Flow Diagram (DFD). Human factors educe issues for design involves addressing internal controls to ensure readability.

## CONCLUSION

It is remarkable that a good anti-phishing system should be able to predict phishing attacks in a reasonable amount of time. Accepting that having a good anti-phishing gadget available at a reasonable time is also necessary for expanding the scope of phishing site detection. The current system merely detects phishing websites using Gradient Boosting Classifier. We achieved 97% detection accuracy using Gradient Boosting Classifier with lowest false positive rate.

## SCOPE FOR FUTURE WORK

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

## REFERENCES

[1] Chengshan Zhang, Steve Sheng, Brad Wardman, Gary Warner, Lorrie Faith Cranor, Jason Hong. Phishing Blacklists: An Empirical Study In: CEAS 2009: Proceedings of the 6th Conference on Email and Anti-Spam, Mountain View, California, USA, July 16-17, 2009.

[2] Andrew Jones, Mahmoud Khonji, Youssef Iraqi, Senior Member A Literature Review on Phishing Detection 2091-2121 in IEEE Communications Surveys and Tutorials, vol. 15, no. 4, 2013. 2013.

[3] Alessandro Acquisti, Idris Adjerid, Rebecca Balebako, Laura Brandimarte, Lorrie Faith Cranor, Saranga Komanduri, Pedro Giovanni Leon, Norman Sadeh, Florian Schaub, Many Understanding and Assisting Users' Online Choices with Nudges for Privacy and Security 50(3), Article No. 44, ACM Computing Surveys, 2017.

[4] Helena Matute, Mara M. Moreno-Fernández, Fernando Blanco, Pablo Garaizar I'm looking for phishers. To combat electronic fraud, Internet users' sensitivity to visual deception indicators should be improved. pp.421-436 in Computers in Human Behavior, Vol.69, 2017.

[5] F.J. Overink, M. Junger, L. Montoya. Preventing social engineering assaults with priming and warnings does not work. pp.75-87 in Computers in Human Behavior, Vol.66, 2017. 2017.

[6] M. El-Alfy, El-Sayed M. Probabilistic Neural Networks and K-Medoids Clustering are used to detect phishing websites. The Computer Journal, 60(12), pp.1745-1759, published in 2017.

[7] Napoleon D. and Praneesh M. “Detection of Brain Tumor using Kernel Induced Possiblistic C-Means Clustering”, volume no.3, issue no.9, pp 436-438, 2013

[8] Shuang Hao, Luca Invernizzi, Yong Fang, Christopher Kruegel, Giovanni Vigna. Cheng Huang, Shuang Hao, Luca Invernizzi, Yong Fang, Christopher Kruegel, Giovanni Vigna. Gossip: Detecting Malicious Domains from Mailing List Discussions Automatically pp. 494-505 in Proceedings of the 2017 ACM Asia Conference on Computer and Communications Security (ASIA CCS 2017), Abu Dhabi, United Arab Emirates, April 2-6, 2017.

[9] Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, Mario Köppen. Frank Vanhoenshoven, Gonzalo Nápoles, Rafael Falcon, Koen Vanhoof, Mario Köppen. Machine learning algorithms are used to detect dangerous URLs. The 2016 IEEE Symposium Series on Computational Intelligence (SSCI 2016) was held on December 6-9, 2016.

[10] Hillary Sanders, Joshua Saxe, Richard Harang, Cody Wild A Deep Learning Approach to Detecting Malicious Web Content in a Fast, Format-Independent Way. pp. 8-14 in Proceedings of the 2018 IEEE Symposium on Security and Privacy Workshops (SPW 2018), San Francisco, CA, USA, August 2.

[11] Jie Wu, Longfei Wu, Xiaojiang Du Phishing Attacks on Mobile Computing Platforms: Effective Defense Schemes 6678-6691 in IEEE Transactions on Vehicular Technology, vol. 65, no. 8, 2016.

[12] Praneesh, M., and R. Annamalai Saravanan. "Deep Stack Neural Networks Based Learning Model for Fault Detection and Classification in Sensor Data." Deep Learning and Edge Computing Solutions for High Performance Computing (2021): 101-110.

[13] A. Kumar, R. S. Umurzoqovich, N. D. Duong, P. Kanani, A. Kuppusamy, M. Praneesh, and M. N. Hieu, ‘‘An intrusion identification and prevention for cloud computing: From the perspective of deep learning,’’ Optik, vol. 270, Nov. 2022, Art. no. 170044.