Phising Website Detection

Saiteja Sreeram
22M0775

Prem Prakash Hansda
22M0818

Jawwad Pathan
22M0741

Abstract

Cyber attack techniques such as phishing, session hijacking, and social engineering are increasingly being used. Malicious users trick victims into entering confidential information by impersonating a legitimate website. Phishing is a social engineering technique that is aimed at a larger target audience with the expectation that a small number of victims will fall victim to it. In this project, we use machine learning classification models to determine whether a website is authentic or a phishing website. We then compare which machine learning model produced the best outcomes.

Keywords: phishing, social engineering , machine learning methods

1 Introduction

With the increase in the use of the internet, people are now more vulnerable to security attacks than ever. Phishing is one such attack carried out by criminals with the purpose of damaging innocent and naïve people. People lose billions of dollars each year as a result of phishing. Phishing is a tactic used by thieves to obtain people's personal information by luring a group of naïve people onto the internet.

Personal information is typically comprised of the username and password for a social media account or a financial account. To deceive people and lure them into their trap, the phisher employs strategies such as spoofed E-mail, SMS, or phishing software. Because most administrative duties relating to the government or some financial institutions can now be completed online via the internet, it is simple to dupe individuals into believing that what they are accessing is authentic.

Phishing emails typically contain some URLs that link to a falsely reproduced website of a real website, which is more often than not a website that requests financial account or social media account information. The email sent to the user will have some sort of signature, such as logos or trademarks of a reputable company. Because the phisher's URL directs the user to the server of a malicious website rather than a legitimate one, current anti-attack solutions such as firewalls and antivirus are ineffective.

> According to the Anti-Phishing Working Group's database, there were a total of 647,592 phishing websites detected..

This research examines the various properties of URLs from benign and phishing websites to provide techniques for dealing with phishing websites. People are looking for a machine learning-based solution to the phishing problem as the use of machine learning in cyber security grows. We will utilize multiple machine learning models to evaluate various aspects of URLs used by both phishing and benign websites. These models are then fine-tuned to achieve the highest possible performance.

2 Machine Learning Methods

We have used different machine learning models in this projects such as-

2.1 Random Forest

The Random Forest ensemble method predicts the outcomes by combining a number of decision tree models. Both classification and regression problems can be solved using the random Forest method. We must average the outcomes from each model before performing regression. Every model in classification problems predicts a class, and the Random Forest Classifier will predict the class that is predicted by the most models. The following is another way to put this: Each model chooses a class, and the Random Classifier foretells the election winner based on the class that receives the most votes.

Since this is an ensemble of trees, each may select a subset of features and construct decision trees; if some trees incorrectly predict the class, they will be countered by other trees, and thus the average or majority of the outputs will provide the best results. As a result, the Random forest model is widely used in a variety of machine-learning classification tasks.

Certain parameters known as hyperparameters can be used to configure each decision tree in the random forest. Hyperparameter tuning refers to the process of configuring hyperparameters.

Hyperparameters in Random Forest:

`n_estimators`: The random forest's tree count. If the number of trees is large, we will get more accurate results because each one predicts a class, whereas if the number of trees is small, it will generalise to a normal decision tree or worse, depending on the other hyperparameters.

criterion: This is the criterion for node splitting. The Gini index and entropy for information gain are included. The node is divided according to the attribute with the greatest information gain when compared to all other attributes.

max_leaf_nodes : This is the maximum depth each decision tree can have in a random forest.

max_leaf_nodes : This is the maximum number of leaf nodes that a tree can have. There are many hyper-parameters that we can tune to get an optimized random forest.

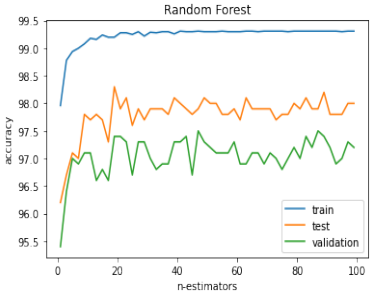


Figure 1: Random forest accuracy

2.2 Support Vector Machine

Support Vector Machine is a supervised machine learning algorithm. The Support Vector Machine model draws a decision boundary that separates all of the data points. Based on the underlying data, this decision boundary may be a line segment or a hyperplane. If the underlying data is linearly separable, the decision boundary is a line segment. If not, the decision boundary is a hyperplane.

There may be multiple decision boundaries between the classes of data points, but we must choose the hyperplane that is equidistant from all of them. Support vectors are the closest points from each class (because they are the contributing factors for the hyperplane). The points that are further away do not contribute to the decision boundary. The distance between the support vectors is referred to as the margin, and the goal of the support vector machine model is to maximise the margin.

The SVM models types: There are two types of SVM models soft-margin and hard-margin SVM models.

Soft margin SVM: This model correctly classifies all the data points even though that makes it complex. This is just an enhancement over Hard-margin SVM. Here the wrongly classified points are called slack variables and they would be given weight and would be considered in the objective function. Thus the objective function becomes the minimization of negative margin and slack variables. The trade-off between these two is obtained using a hyperparameter(C). If the value of C is zero, then this would become hard-margin SVM. If the parameter is low, then the model would mainly focus on maximizing the margin and if it's high, then the model would focus on correctly classifying the samples.

Hard-margin SVM: Hard-margin SVM: Here the model gives the lenience for some of the wrong data points. That is some of the data points of one class may lie in another class thereby making the model simpler. This is due to some data points might be outliers or may not be of much importance. So, the model doesn't consider these and creates a hyperplane since this model ease out some points, this is called as hard-margin

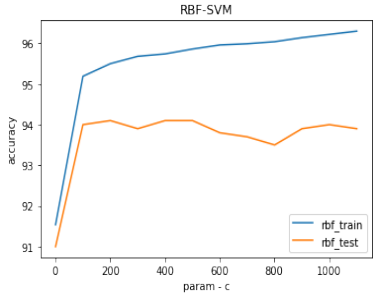


Figure 2: RBF-SVM accuracy vs param c

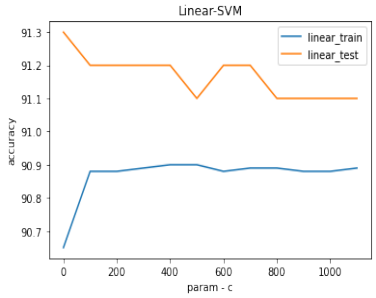


Figure 3: Linear-SVM accuracy vs param c

SVM makes use of kernels. The data in the figure on the right is non-linearly separable, but by applying a kernel, it becomes linearly separable and thus easily classifiable. There are numerous kernel types, such as linear and Gaussian. A kernel function can be validated using a variety of methods. Every kernel adheres to kernel composition rules known as the sum rule, scaling rule, and product rule.

Each node will compare the attribute in this case. Each branch will lead to one of the attribute's possible values. Then another node splits it, and the branches lead to one of the node's values. This splitting continues until we reach the leaf node. The leaf would be homogeneous, which means that all of the tuples in the leaf node belong to the same class.

2.3 Decison Tree

Each node will compare the attribute here. Each branch will lead to one of the possible values for that attribute. Then it is split by another node, and the branches lead to one of the node's values. This process is repeated until we reach the leaf node. The leaf would be homogeneous, which means that all of the tuples in the leaf node are of the same class.

Thus, when testing the model with the validation dataset, we have a tuple that is checked across the root node for its value and traverses along the branch that has the same value as the tuple, and then the tuple is checked at level 2 node for its value and traverses along the branch with the same value and traverses until it reaches the leaf node. The predicted class of the leaf node is compared to the class of the data tuple, and the model's accuracy is calculated. The model's hyperparameters are then tuned to produce the best tree.

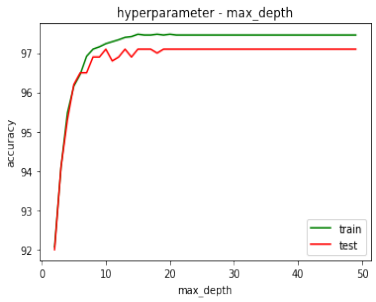


Figure 4: accuracy vs max depth

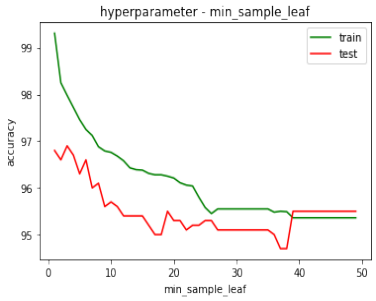


Figure 5: accuracy vs minimum sample leaf

Hyperparameter of Decision Tree :

`criterion`: This is the criterion for selecting a splitting node like the node which gives more information gain that will be used as a splitting node.

`max_depth`: The decision tree has reached its maximum depth. If the data does not become homogeneous even after reaching the leaf nodes, the class assigned to those data points is that which was in the majority of the training data points. If the maximum depth is set too high, the model overfits the data and produces poor results on test data.

`min_samples`: This is the smallest number of data points that should exist at a node before splitting. If the number of nodes at a node is less than this threshold, no split will occur on that node.

`max_leaf_nodes`: This is the maximum number of leaf nodes that can exist in a tree.

Many more hyperparameters can be tuned on the validation dataset to produce a good decision tree, which can then be used on the test dataset to predict values for new tuples or data points.

2.4 Multi-layer Perceptron

:

Multi-layer perceptron is one of the neural network models. A neural network model consists of certain nodes called neurons and there would be connections between neurons. These neurons form a layered structure where there would be connections between neurons of one layer to neurons of another layer. Each input connection is given some weight and we calculate the weighted sum of neurons.

This is to be used as an input to an activation function, and the neuron will output as a result. The outputs from all of the neurons in one layer are considered as inputs to the neurons in the following layer, and the same thing occurs at the neurons in that layer as well until we reach the output layer. Based on the kind of need the model has, a separate activation function will be present in the output layer. Softmax classification is used to forecast the data point that belongs to a class with a high probability when there are several neurons in the output layer and we need to predict a class.

The parameters of a neural network model are these weights and biases, which are also added by each neuron. To improve the model, we identify the gradients of neurons with regard to biases and eights.

The backpropagation method will be used to calculate the gradients. Overfitting is a major risk for multi-layer perceptron. Many techniques have been used to prevent overfitting. The dropout approach is one technique. In this system, each neuron receives a probability for each data point, and is only activated if the probability exceeds a certain threshold. Otherwise, the neuron is dormant. A neuron is said to be active if it processes input and produces output, and inactive if it rejects input and does not produce output.

So, we evaluate the likelihood for each data point, and as a result, some neurons will be active for some data points, and for some other data points, neurons that were inactive for the previous data points will become active. Thus, a collection of several multi-layer perceptron models is effectively created. The accuracy would therefore be good.

Hyperparameter of Multi-layer perceptron:

`hidden_layer_size`: This gives the measure of number of layers in the model. More the number of layers, more complex the model becomes.

`activation_function`: Non-linear functions like sigmoid, tan, ReLU(Rectified Linear Unit) can be used.

`solver`: this is the optimization technique to be used . We can use adam, adagrad SGD with momentum.

`alpha`: This is the co-efficient of L2 regularized model. If the alpha value is zero, then the model becomes unregularized model.

`learning_rate`: learning rate can be any of 'constant', 'adaptive' and 'invscaling'.

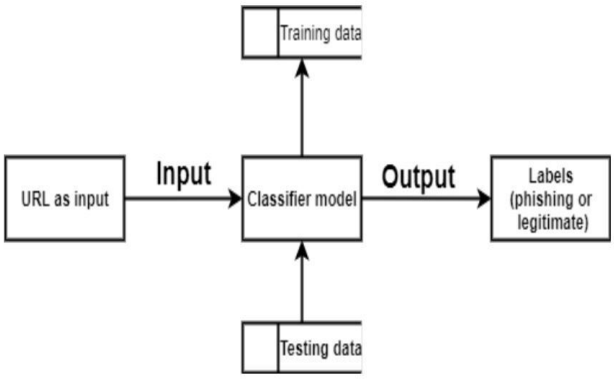
There are many more hyperparameters that can be used to get an optimized model.

3 System Design

`Data Flow Diagram`

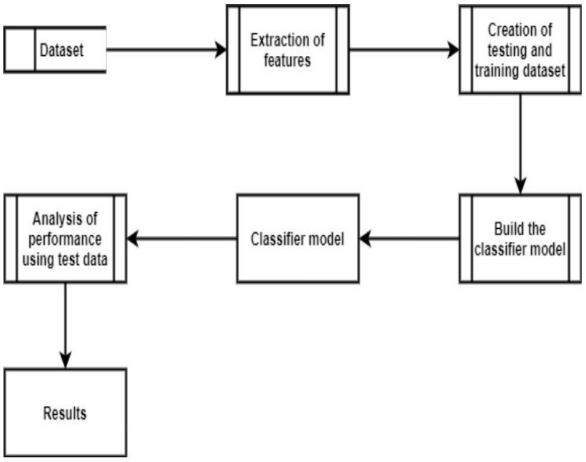
The graphical representation data flow in the system is called data flow diagram. It gives the idea about all the events that are happening in the system from input to output generation. There are different levels present namely level 0,1,2 which explains the system in different depths of details.

3.1 Level 0



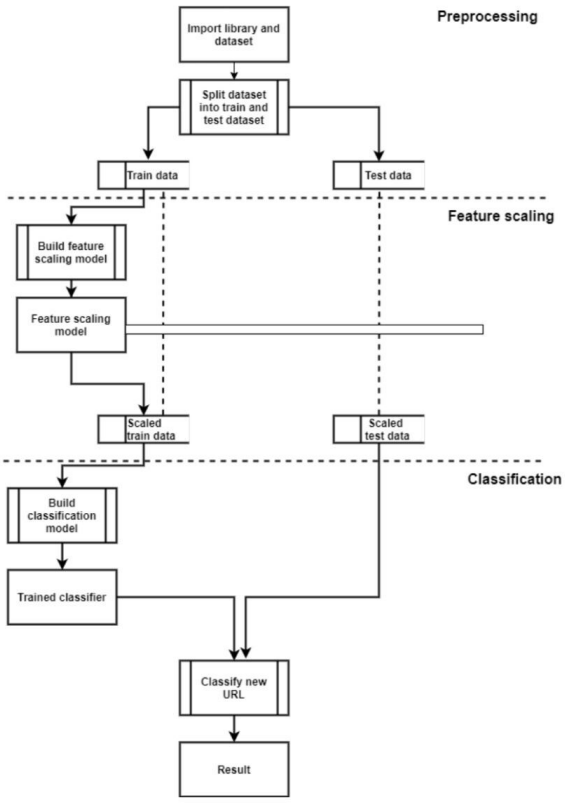
It is the most simple representation of the system. It is also called context diagram. It helps us to get a high level overview of the system.It is the most simple representation of the system.

3.2 Level 1



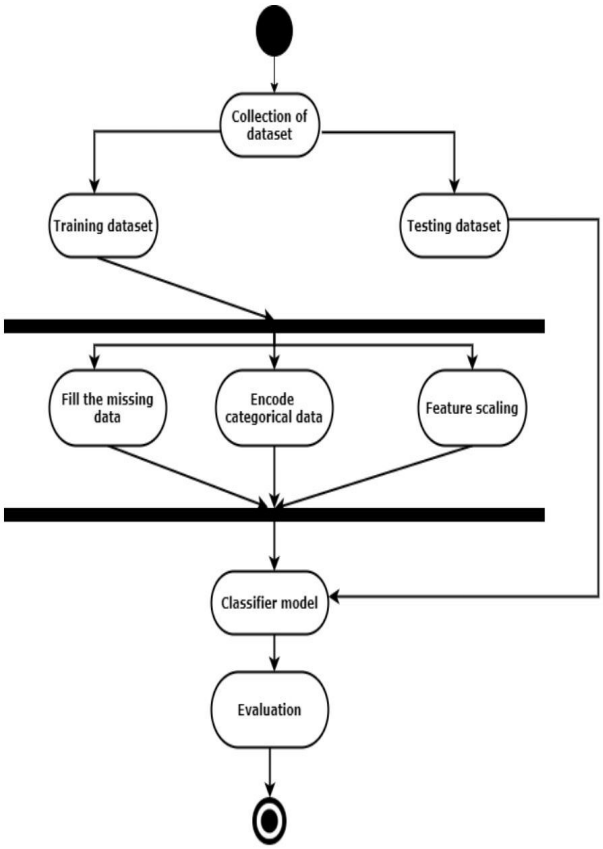
It gives a more detailed explanation of the system than level 0. The high level diagram in level 0 is broken into different subprocesses. The steps such as splitting the dataset, feature selection, building the classifier.

3.3 Level 2



A more detailed explanation of steps in level 1 is given by level 2. More amount of text is present in the diagram to explain the functions of the system in detail.

UML Diagram



The control flow of the system from input to output is given by UML activity diagram. It shows various paths that exist in the system.

4 Implementation

Collection of dataset: The first step of implementation was the selection of the dataset. We selected the dataset from Kaggle. Our dataset consists of 10000 rows and 50 columns. Some of the features of the dataset are listed below –

1. NumDots - Number of character '.' in URL
2. UrlLength - The length of the URL
3. NoHttps - Check if there exists an HTTPS in the website URL
4. NumAmpersand - Number of the character ''
5. PctExtHyperlink - The percentage of external hyperlinks in the HTML source code of the website

Preprocessing: The preprocessing step usually requires the missing values to be amputated by different methods but in our case, the dataset didn't contain any missing values and was clean from the start. Most of the features in our dataset are binary and some are numeric but the difference in scaling is not significant so we didn't need to do normalization of data as well.

Feature selection: The process of selecting the important features in the dataset to reduce the running time complexity of the model is known as feature selection. We are implementing feature selection using two techniques which are as follows-

1. Correlation matrix
2. Forward Featre Selection

In the first method, we are finding the correlation of each of the features with the class label and then select the features with the highest correlation values. The correlation matrix is for our dataset given below- In the second method, we are doing Forward Feature Selection which is an iterative technique in which we start from 0 features and keep adding features which improve the accuracy of our model. The graph for accuracy vs number of features is given below.

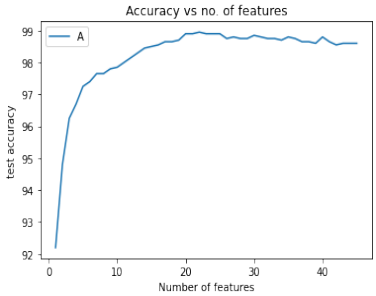


Figure 6: accuracy vs number of features

Splitting: It is the process of splitting the data into train, test and cross-validation sets. The ratio of the split is 80

Models:Applying different machine learning models on the train set and tuning the hyperparameters using the cross-validation set and testing the accuracy on the test set. Compare the test accuracy of different models and select the best one.

5 Results

Model	FFS	Correlation
Random Forest	98.40	98.00
Decision Tree	97.45	97.00
Multi Layer Perceptron	96.69	96.81
SVM	94.85	94.40

Table 1: F1 score comparison of different machine learning models.

6 Conclusion

Online businesses, exchanges without cash, Paperless tickets and so on are being done on a large scale. Phishing is an obstacle to such advancements in technology. We must understand that it is important to spread awareness about attacks such as phishing. People must be made aware of the security measures to take to avoid being exploited by such attacks. Every user must make sure not to blindly just believe any message they receive and disclose some sensitive information. People must have some idea about determining whether an URL is legitimate or not. The advancement of AI in cybersecurity is commendable. We are using AI in this project to classify whether the site is phishing or not. Out of the 4 algorithms that were used, Random Forest gave the best result with an accuracy of 98.40.