PROJECT REPORT ON
WEBSITES PHISHING DETECTION
USING DATA MINING

Khushboo Jha
Khushboo_jha@rutgers.edu

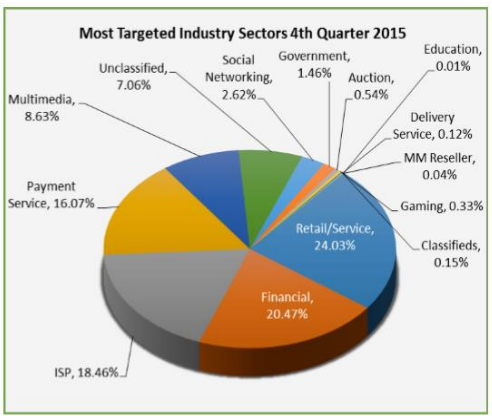
Contents

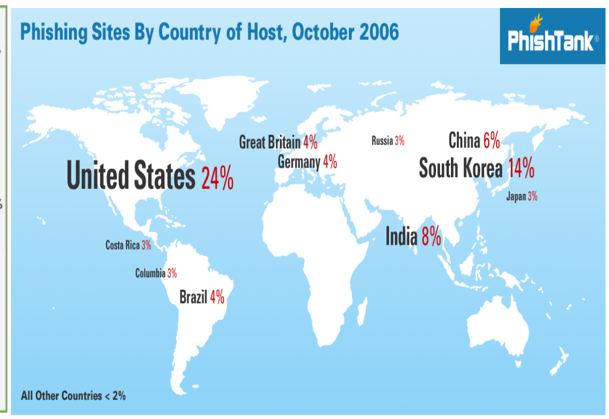
PHISHING OVERVIEW

Phishing is a criminal mechanism employing both social engineering and technical subterfuge to steal consumers' personal identity data and financial account credentials.

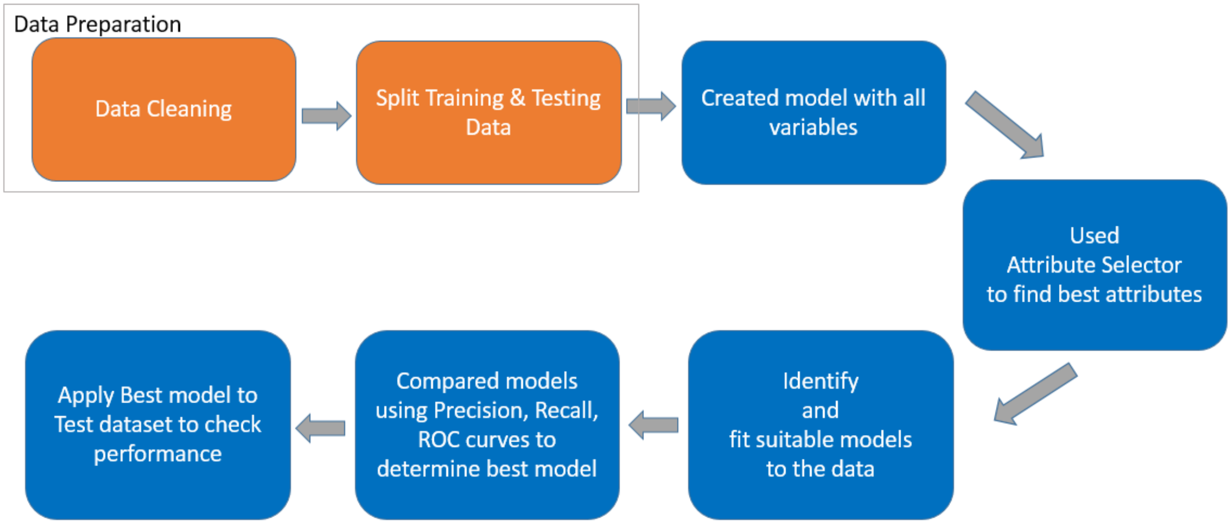
Phishing refers to the process where a targeted individual is contacted by email or telephone by someone posing as a legitimate institution to lure the individual into providing sensitive information such as banking information, credit card details, and passwords. The personal information is then used to access the individual's account and can result in identity theft and financial loss. It is a global security threat. According to Consumer Reports, the cost of phishing is nearly $500 million per year in the United States alone.

United States most targeted country, Retail/Service & Financial are the most targeted sectors





MODEL BUILDING PROCEDURE



Above diagram depicts our model building procedure .We start with cleaning the data by removing missing values from our data .We then try to create C4.5 model using all the 31 attributes of our data. After applying the model we used attribute selector feature in WEKA to select the best attributes from our data which are least correlated. These attributes matched with our selection of attributes based on our domain knowledge by referring different research papers. Next step is to select the best fit models .We used 4 models namely C4.5, Random Forests, Naïve Bayes and SVM. We compared the different models with their confusion matrix and ROC curves. We then apply the best model on the test data and evaluate the results.

DATA DESCRIPTION

The dataset consists of 31 attributes and 2670 instances and can be classified into 2 main categories

<u>URL-based features:</u> These features Extracted from the webpage's URL and its meta-data. Examples of feature creation

$$\text{IF}\begin{cases}\text{Url Having @ Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate}\end{cases} \quad \text{IF}\begin{cases}\text{Domain Name Part Includes } (-) \text{ Symbol} \rightarrow \text{Phishing} \\ \text{Otherwise} \rightarrow \text{Legitimate}\end{cases}$$

<u>Content-based features:</u> These features are extracted from the source code of Web page. Examples of feature creation

$$\text{IF}\begin{cases}\text{Use https and Issuer Is Trusted and Age of Certificate} \geq 1 \text{ Years} \rightarrow \text{Legitimate} \\ \text{Using https and Issuer Is Not Trusted} \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing}\end{cases}$$

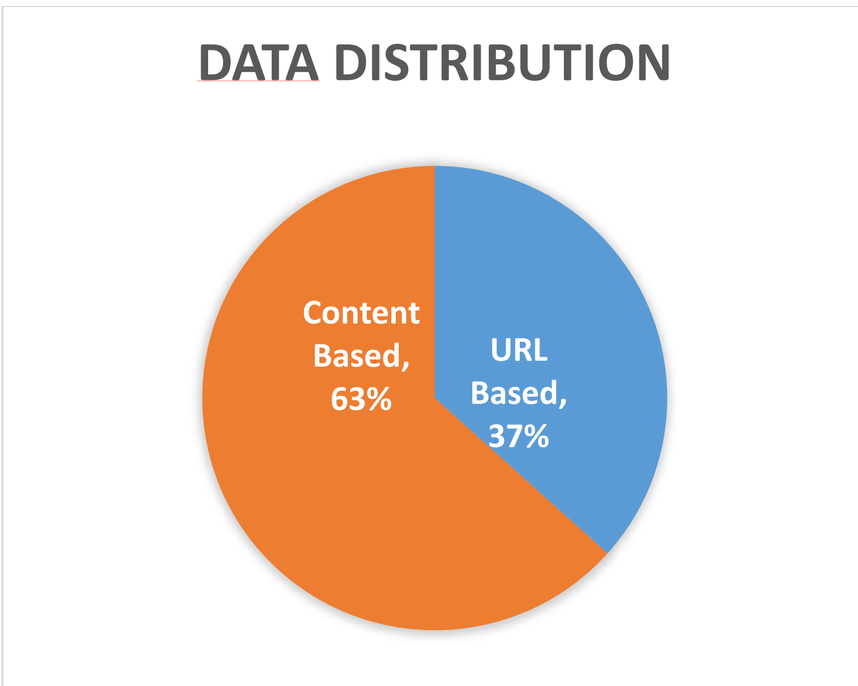
$$\text{IF}\begin{cases}\text{Website Rank} < 100{,}000 \rightarrow \text{Legitimate} \\ \text{Website Rank} > 100{,}000 \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phish}\end{cases}$$

Our values of features are

1 – Phishing
0 – Suspicious
1 – Legitimate

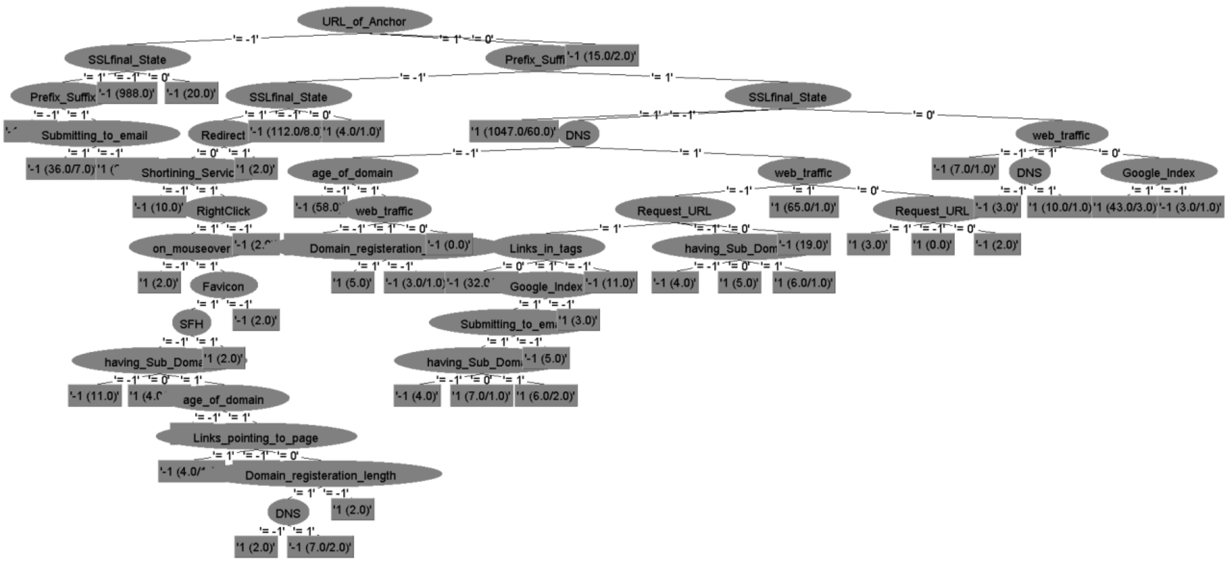
DATA PREPROCESSING:

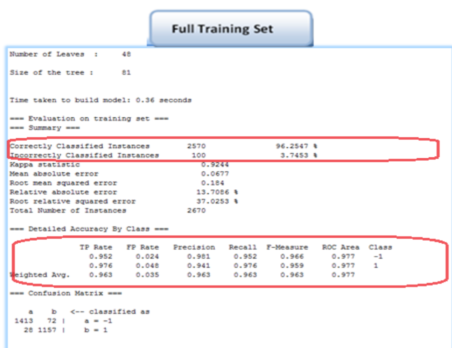
- This consisted mainly of removal of missing values. We did not have outliers in our dataset. After preprocessing our training dataset consisted of 31 attributes and 2670 instances. Testing dataset contains 31 attributes and 2456 instances



Model Building

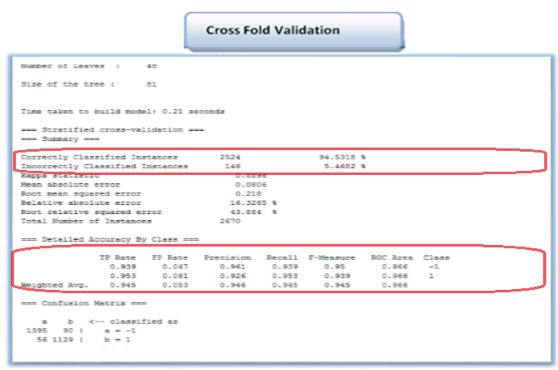
Our first step in model building consisted of using the model C4.5 using all 31 attributes. The results are shown below. The next step is attribute selection and then using the selected attributes to build the model.





Full Training Set

```
Number of Leaves  :     48

Size of the tree :      81


Time taken to build model: 0.36 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        2570               96.2547 %
Incorrectly Classified Instances       100                3.7453 %
Kappa statistic                          0.9244
Mean absolute error                      0.0677
Root mean squared error                  0.184
Relative absolute error                 13.7086 %
Root relative squared error             37.0253 %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.952     0.024       0.981     0.952     0.966      0.977    -1
                 0.976     0.048       0.941     0.976     0.959      0.977    1
Weighted Avg.    0.963     0.035       0.963     0.963     0.963      0.977

=== Confusion Matrix ===

    a    b   <-- classified as
 1413   72 |    a = -1
   28 1157 |    b = 1
```



Cross Fold Validation

```
Number of Leaves  :     48

Size of the tree :      81


Time taken to build model: 0.21 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2524               94.5318 %
Incorrectly Classified Instances       146                5.4682 %
Kappa statistic                          0.8896
Mean absolute error                      0.0806
Root mean squared error                  0.218
Relative absolute error                 16.3265 %
Root relative squared error             43.884  %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.939     0.047       0.961     0.939     0.95       0.966    -1
                 0.953     0.061       0.926     0.953     0.939      0.966    1
Weighted Avg.    0.945     0.053       0.946     0.945     0.945      0.966

=== Confusion Matrix ===

    a    b   <-- classified as
 1395   90 |    a = -1
   56 1129 |    b = 1
```

Summary	Error Rate	TP Rate	FP Rate	Precision	Recall	F-Measure
Full Training Set	3.7453	0.963	0.035	0.963	0.963	0.963
Cross fold Validation	5.4682	0.945	0.053	0.946	0.945	0.945

FEATURE SELECTION

Using the CfsSubsetEval function (Attribute Evaluator)

- Considering the individual predictive ability of each feature along with the degree of redundancy between them.
- Check subsets of features that are highly correlated with the class while having low inter correlation are preferred.

Using different Search methods we get the same best attributes

- **Greedy Stepwise:** Performs a greedy forward or backward search through the space of attribute subsets. May start with no/all attributes or from an arbitrary point in the space. *Stops when the addition/deletion of any remaining attributes results in a decrease in evaluation.* Can also produce a ranked list of attributes by traversing the space from one side to the other and recording the order that attributes are selected.
- **Best First:** Searches the space of attribute subsets by greedy hill climbing augmented with a backtracking facility. Setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. Best first may start with the empty set of attributes and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions (by considering all possible single attribute additions and deletions at a given point).
- **Rank Search:** Uses an attribute/subset evaluator to rank all attributes. If a subset evaluator is specified, then a forward selection search is used to generate a ranked list. From the ranked list of attributes, subsets of increasing size are evaluated, i.e. the best attribute, the best attribute plus the next best attribute, etc.... The best attribute set is reported. Rank Search is linear in the number of attributes if a simple attribute evaluator is used such as GainRatioAttributeEval.

Selected attributes were the **least correlated** and produced the best results for the models

Variable Selection by Different Methods

Greedy Stepwise

- Prefix_Suffix
- having_Sub_Domain
- URL_of_Anchor
- Request_URL
- SSLfinal_State
- Links_in_tags
- SFH

- age_of_domain

Best First

- Prefix_Suffix
- having_Sub_Domain
- URL_of_Anchor
- Request_URL
- SSLfinal_State
- Links_in_tags
- SFH
- age_of_domain

Rank Search

- Prefix_Suffix
- having_Sub_Domain
- URL_of_Anchor
- Request_URL
- SSLfinal_State
- Links_in_tags
- SFH
- age_of_domain

Correlation Matrix of Selected Variables

Attributes	Prefix_Suffix	having_Sub_Domain	URL_of_Anchor	Request_URL	SSLfinal_State	Links_in_tags	SFH	age_of_domain
Prefix_Suffix	1.000							
having_Sub_Domain	0.206	1.000						
URL_of_Anchor	0.552	0.170	1.000					
Request_URL	-0.033	-0.047	0.008	1.000				
SSLfinal_State	0.482	0.235	0.636	-0.046	1.000			
Links_in_tags	0.137	0.069	0.149	-0.040	0.134	1.000		
SFH	-0.057	-0.048	-0.034	0.014	-0.060	0.019	1.000	
age_of_domain	0.129	0.200	0.198	0.045	0.249	-0.028	-0.035	1.000

What every attribute mean:

Prefix_Suffix: 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 $\begin{cases} \text{Domain Name Part Includes } (-) \text{ Symbol } \rightarrow \text{ Phishing} \\ \text{Otherwise } \rightarrow \text{ Legitimate} \end{cases}$

having_Sub_Domain:

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 have to omit the (www.) from the URL which is in fact a sub domain in itself. Then, we have to 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 $\begin{cases} \text{Dots In Domain Part} = 1 \rightarrow \text{ Legitimate} \\ \text{Dots In Domain Part} = 2 \rightarrow \text{ Suspicious} \\ \text{Otherwise} \rightarrow \text{ Phishing} \end{cases}$

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:

If the <a> tags and the website have different domain names. This is similar to request URL feature.

If the anchor does not link to any webpage, e.g.:

<a href="#">

<a href="#content">

<a href="#skip">

<a href="JavaScript ::void(0)">

$$\text{Rule: IF}\begin{cases} \text{\% of URL Of Anchor } < 31\% \rightarrow \textit{Legitimate} \\ \text{\% of URL Of Anchor } \geq 31\% \text{ And} \leq 67\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

Request_URL:

Request URL examines whether the external objects contained within a webpage 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.

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

SSLfinal_State: This indicates security of website.

Links_in_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

$$\begin{cases} \text{\% of Links in " } < \text{Meta} > \text{"," } < \text{Script} > \text{" and " } < \text{Link>" } < 17\% \rightarrow \text{Legitimate} \\ \text{\% of Links in } < \text{Meta} > \text{"," } < \text{Script} > \text{" and " } < \text{Link>" } \geq 17\% \text{ And} \leq 81\% \rightarrow \text{Suspicious} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

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.

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

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.

$$\text{Rule: IF}\begin{cases} \text{Age Of Domain} \geq 6 \text{ months} \rightarrow \text{Legitimate} \\ \text{Otherwise} \rightarrow \text{Phishing} \end{cases}$$

ATTRIBUTES VISUALISATION:

- The picture depicts the capability of individual attributes in the overall classification of phishing websites
- Identifying legitimate sites as phishy and phishy sites as legitimate contributes to the error rate (depicted in relatively smaller bubble)
- Big bubble depicts 'True Positive' & 'True Negative', small bubble helps in finding the error rate. We try to keep minimum smaller bubbles to achieve good results.

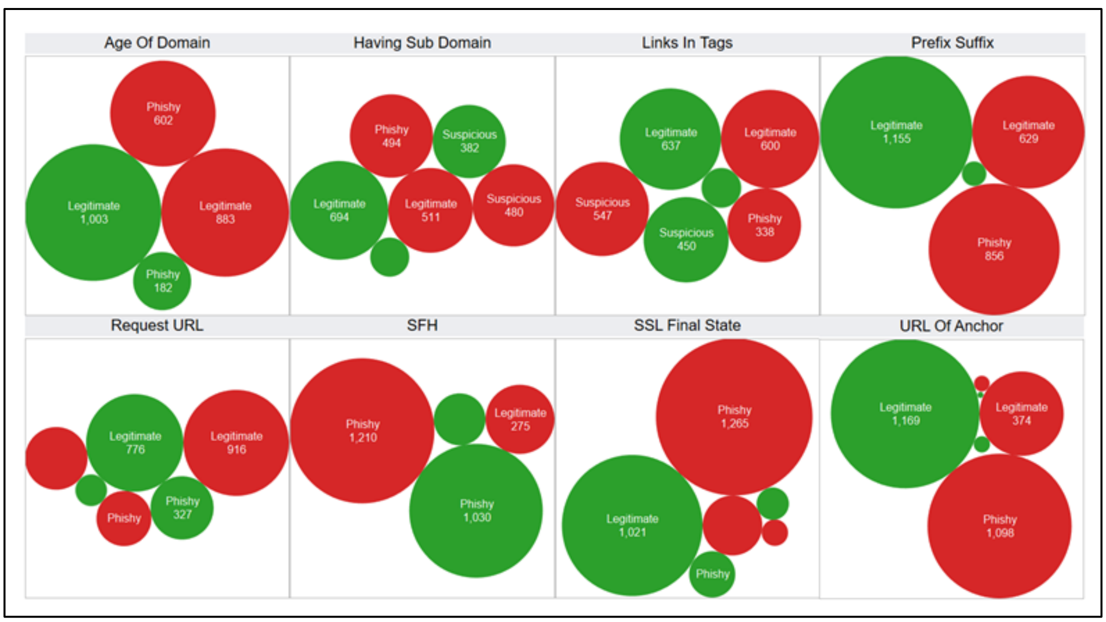


Figure 1:Visualization of capabilities of individual attributes in determining class labels

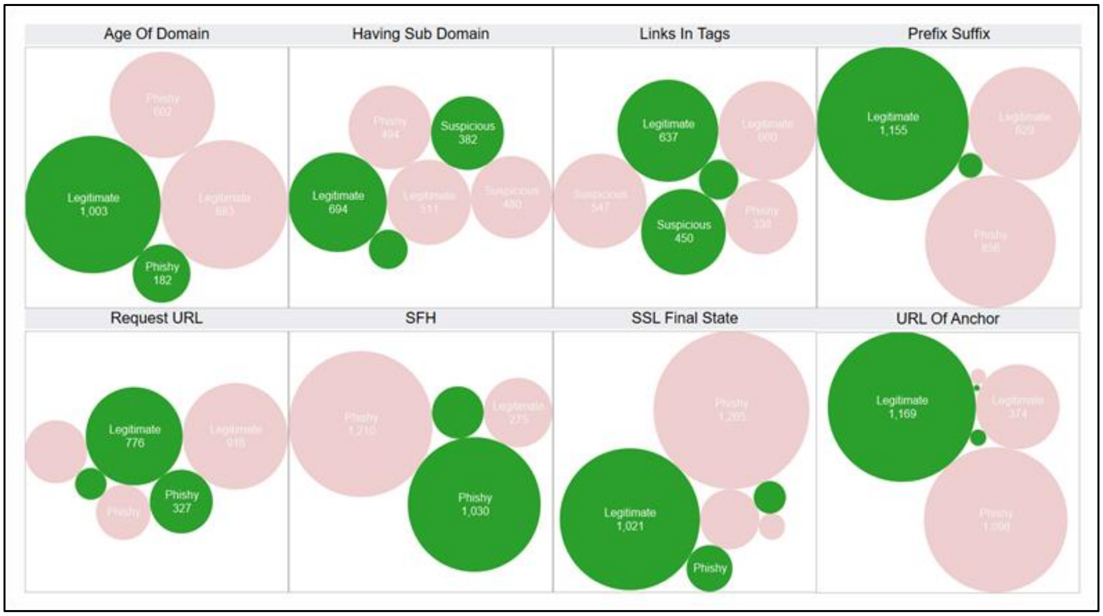


Figure 2: Visualization of capabilities of individual attributes in determining class label: Legitimate

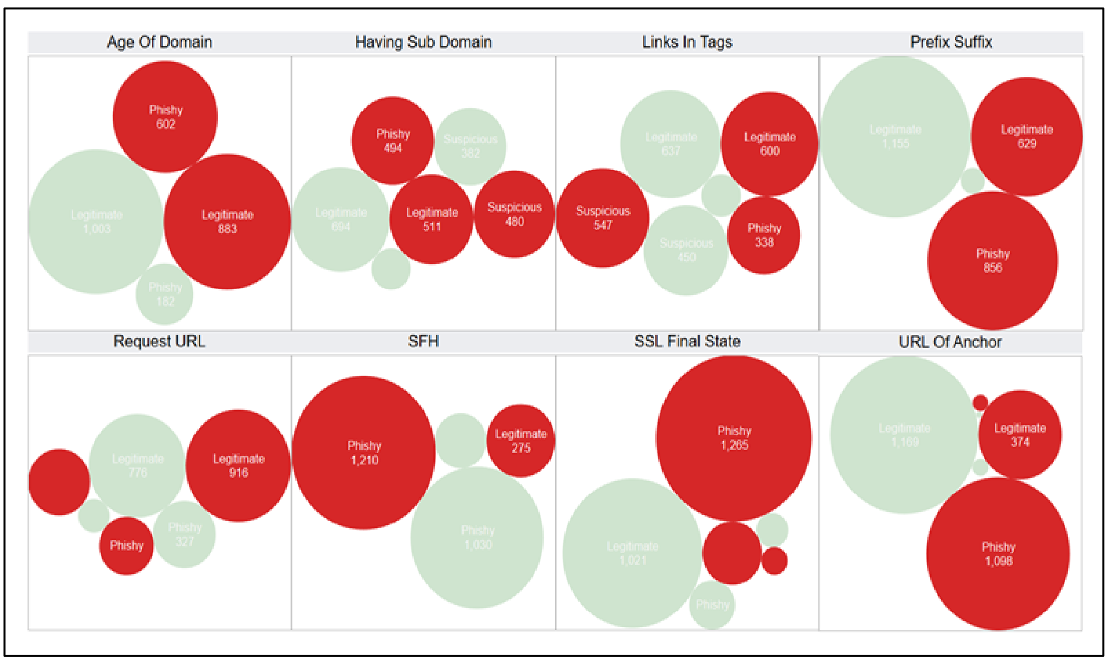


Figure 3: Visualization of capabilities of individual attributes in determining class label : Phishy

MODEL SELECTION

Based on the data we have chosen to fit the following classification models:

C4.5

- Indirect Method: Extract rules from other classification models (e.g. decision trees, neural networks, SVM, etc).
- Rule based Classifier that is similar to the data
- Good for implementation as it is easy to interpret, generate and can classify new instances rapidly
- Rules that predict the same class are grouped together into the same subset

Random Forest

- Tree / Rule based Classifier that is similar to the data
- Builds multiple trees and uses a voting system to get the end result => very thorough examination
- Provides a very good estimate of the generalization error of a classifier
  Random forest (RF) is an ensemble learning classification and regression method suitable for handling problems involving grouping of data into classes. The algorithm was developed by Breiman and Cutler. In RF, prediction is achieved using decision trees. During the training phase, a number of decision trees are constructed (as defined by the programmer) which are then used for the class prediction; this is achieved by considering the voted classes of all the individual trees and the class with the highest vote is considered to be the output.

Support Vector Machine (SVM)

- Good for classification of two groups and performs very well for high-dimensional data
- The results are stable, reproducible and independent of specific optimization algorithms
- The results are repeatable when the parameter is fixed

Naïve Bayes

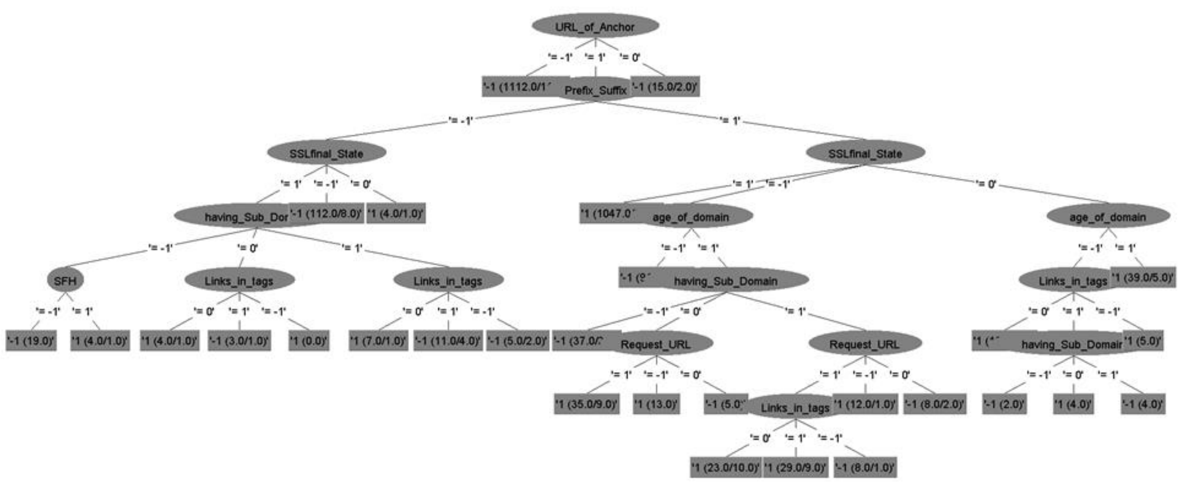
- Given a set of classified training samples, the application can learn from these samples to predict the class of a new unclassified sample.

MODEL EVALUATION

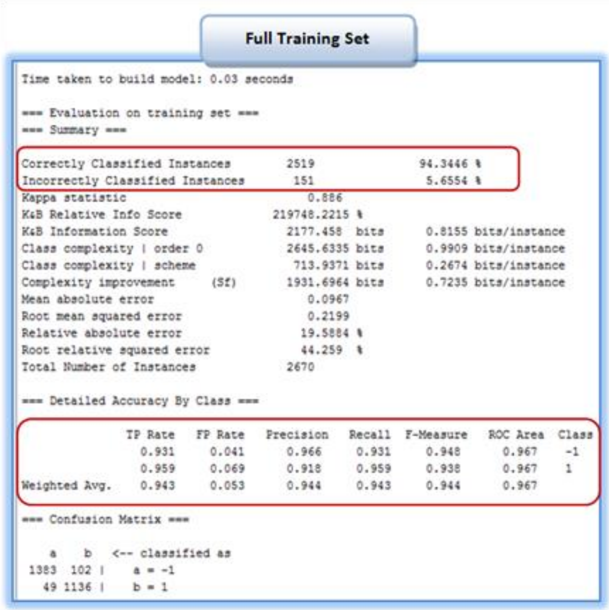
Using our training dataset, the selected models are then built using Weka 3.8 and results are then visualized and compared. The models are built using two test options namely full training set and 10-fold cross validation. This is because it is often necessary to perform validation on our training sets to achieve greedy performance measures.

C4.5

The decision tree obtained using C4.5 model (J48 as called in Weka) using our selected attributes is as below:

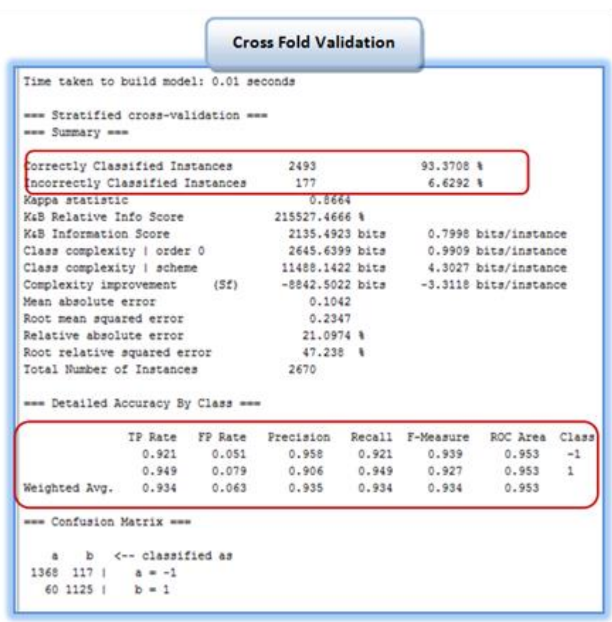


The results of C4.5 model is summarized below:



Full Training Set

```
Time taken to build model: 0.03 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        2519               94.3446 %
Incorrectly Classified Instances       151                5.6554 %
Kappa statistic                          0.886
K&B Relative Info Score             219748.2215 %
K&B Information Score                 2177.458  bits      0.8155 bits/instance
Class complexity | order 0            2645.6335 bits      0.9909 bits/instance
Class complexity | scheme              713.9371 bits      0.2674 bits/instance
Complexity improvement     (Sf)       1931.6964 bits      0.7235 bits/instance
Mean absolute error                      0.0967
Root mean squared error                  0.2199
Relative absolute error                 19.5884 %
Root relative squared error             44.259  %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.931     0.041     0.966     0.931     0.948      0.967    -1
                 0.959     0.069     0.918     0.959     0.938      0.967    1
Weighted Avg.    0.943     0.053     0.944     0.943     0.944      0.967

=== Confusion Matrix ===

    a    b   <-- classified as
 1383  102 |    a = -1
   49 1136 |    b = 1
```



Cross Fold Validation

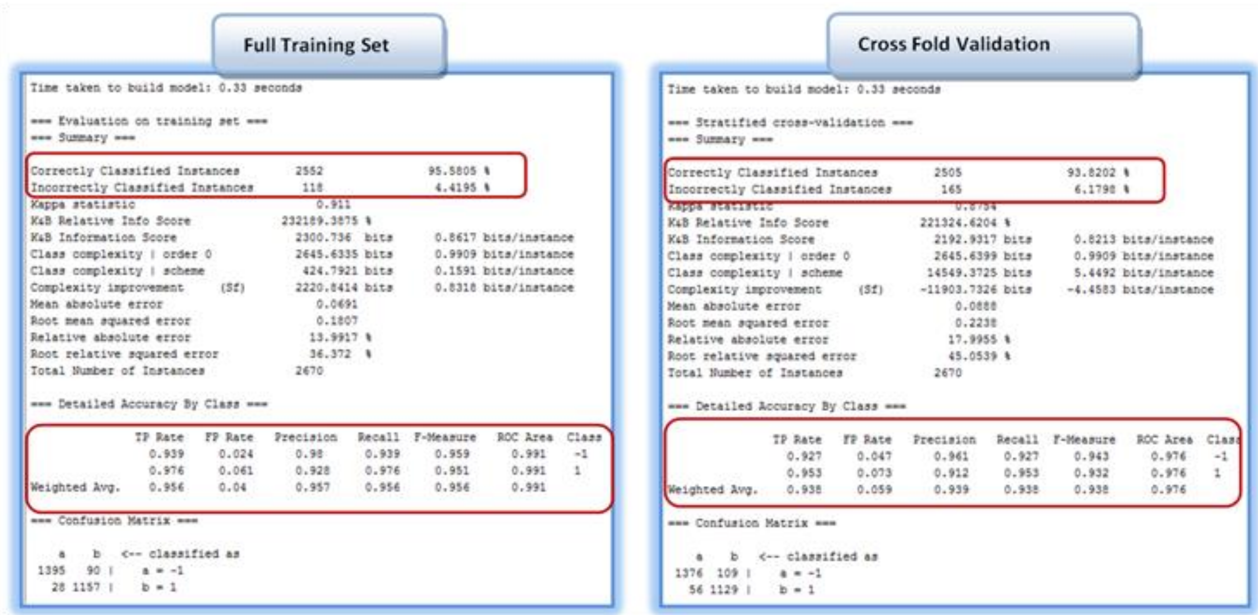
```
Time taken to build model: 0.01 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2493               93.3708 %
Incorrectly Classified Instances       177                6.6292 %
Kappa statistic                          0.8664
K&B Relative Info Score             215527.4666 %
K&B Information Score                 2135.4923 bits      0.7998 bits/instance
Class complexity | order 0            2645.6399 bits      0.9909 bits/instance
Class complexity | scheme            11488.1422 bits      4.3027 bits/instance
Complexity improvement     (Sf)      -8842.5022 bits     -3.3118 bits/instance
Mean absolute error                      0.1042
Root mean squared error                  0.2347
Relative absolute error                 21.0974 %
Root relative squared error             47.238  %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.921     0.051     0.958     0.921     0.939      0.953    -1
                 0.949     0.079     0.906     0.949     0.927      0.953    1
Weighted Avg.    0.934     0.063     0.935     0.934     0.934      0.953

=== Confusion Matrix ===

    a    b   <-- classified as
 1368  117 |    a = -1
   60 1125 |    b = 1
```

Table 1: Summary of test results - C4.5 model

Summary	Error Rate	TP Rate	FP Rate	Precision	Recall	F-Measure
Full Training Set	5.6554 %	0.928	0.068	0.93	0.928	0.929
Cross fold Validation	6.6292 %	0.934	0.063	0.935	0.934	0.934

Random Forest

The Random Forest are built using 100 trees and out of bag error is 0.0637. The results of output is summarized below:



Full Training Set

```
Time taken to build model: 0.33 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        2552               95.5805 %
Incorrectly Classified Instances       118                4.4195 %
Kappa statistic                          0.911
K&B Relative Info Score             232189.3875 %
K&B Information Score                 2300.736  bits      0.8617 bits/instance
Class complexity | order 0            2645.6335 bits      0.9909 bits/instance
Class complexity | scheme              424.7921 bits      0.1591 bits/instance
Complexity improvement     (Sf)       2220.8414 bits      0.8318 bits/instance
Mean absolute error                      0.0691
Root mean squared error                  0.1807
Relative absolute error                 13.9917 %
Root relative squared error             36.372  %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.939     0.024      0.98      0.939     0.959      0.991    -1
                 0.976     0.061      0.928     0.976     0.951      0.991    1
Weighted Avg.    0.956     0.04       0.957     0.956     0.956      0.991

=== Confusion Matrix ===

    a    b   <-- classified as
 1395   90 |    a = -1
   28 1157 |    b = 1
```

Cross Fold Validation

```
Time taken to build model: 0.33 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2505               93.8202 %
Incorrectly Classified Instances       165                6.1798 %
Kappa statistic                          [illegible]
K&B Relative Info Score             221324.6204 %
K&B Information Score                 2192.9317 bits      0.8213 bits/instance
Class complexity | order 0            2645.6399 bits      0.9909 bits/instance
Class complexity | scheme            14549.3725 bits      5.4492 bits/instance
Complexity improvement     (Sf)     -11903.7326 bits     -4.4583 bits/instance
Mean absolute error                      0.0888
Root mean squared error                  0.2238
Relative absolute error                 17.9955 %
Root relative squared error             45.0539 %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.927     0.047      0.961     0.927     0.943      0.976    -1
                 0.953     0.073      0.912     0.953     0.932      0.976    1
Weighted Avg.    0.938     0.059      0.939     0.938     0.938      0.976

=== Confusion Matrix ===

    a    b   <-- classified as
 1376  109 |    a = -1
   56 1129 |    b = 1
```

Table 2: Summary of test results – Random Forest Model

Summary	Error Rate	TP Rate	FP Rate	Precision	Recall	F-Measure
Full Training Set	4.4195 %	0.956	0.040	0.957	0.956	0.956
Cross fold Validation	4.1798 %	0.938	0.059	0.939	0.938	0.938

Support Vector Machine (SVM)

The support vector machine distinguishes two class labels by creating a well-defined boundary between two classes by maximizing the distance between them. The model is using a linear kernel K (x, y) function. The output model with normalized support vectors as displayed in weka is as follows:

```
Kernel used:
  Linear Kernel: K(x,y) = <x,y>

Classifier for classes: -1, 1

BinarySMO

Machine linear: showing attribute weights, not support vectors.

         1.999  * (normalized) Prefix_Suffix
 +      -0.6662 * (normalized) having_Sub_Domain=-1
 +       0.3329 * (normalized) having_Sub_Domain=0
 +       0.3333 * (normalized) having_Sub_Domain=1
 +      -1.3334 * (normalized) URL_of_Anchor=-1
 +       1.6662 * (normalized) URL_of_Anchor=1
 +      -0.3328 * (normalized) URL_of_Anchor=0
 +      -0.0003 * (normalized) Request_URL=1
 +       0.9993 * (normalized) Request_URL=-1
 +      -0.9989 * (normalized) Request_URL=0
 +       0.6665 * (normalized) SSLfinal_State=1
 +      -1.333  * (normalized) SSLfinal_State=-1
 +       0.6665 * (normalized) SSLfinal_State=0
 +       0      * (normalized) Links_in_tags=0
 +       0.0005 * (normalized) Links_in_tags=1
 +      -0.0004 * (normalized) Links_in_tags=-1
 +      -0.0003 * (normalized) SFH
 +       0.9991 * (normalized) age_of_domain
 -       3.6644

Number of kernel evaluations: 1028382 (75.956% cached)
```

Full Training Set

```
Time taken to build model: 0.84 seconds

=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        2467               92.397 %
Incorrectly Classified Instances       203                7.603 %
Kappa statistic                          0.8462
K&B Relative Info Score             225648.0053 %
K&B Information Score                 2235.9182 bits      0.8374 bits/instance
Class complexity | order 0            2645.6335 bits      0.9909 bits/instance
Class complexity | scheme           218022      bits     81.6562 bits/instance
Complexity improvement     (Sf)    -215376.3665 bits    -80.6653 bits/instance
Mean absolute error                      0.076
Root mean squared error                  0.2757
Relative absolute error                 15.4003 %
Root relative squared error             55.4985 %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.927     0.08      0.935      0.927     0.931      0.924    -1
                 0.92      0.073     0.91       0.92      0.915      0.924     1
Weighted Avg.    0.924     0.077     0.924      0.924     0.924      0.924

=== Confusion Matrix ===

    a    b   <-- classified as
 1377  108 |    a = -1
   95 1090 |    b = 1
```

Cross Fold Validation

```
Time taken to build model: 0.61 seconds

=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2469               92.4719 %
Incorrectly Classified Instances       201                7.5281 %
Kappa statistic                          0.8478
K&B Relative Info Score             226071.098  %
K&B Information Score                 2239.9609 bits      0.8389 bits/instance
Class complexity | order 0            2645.6399 bits      0.9909 bits/instance
Class complexity | scheme           215874      bits     80.8517 bits/instance
Complexity improvement     (Sf)    -213228.3601 bits    -79.8608 bits/instance
Mean absolute error                      0.0753
Root mean squared error                  0.2744
Relative absolute error                 15.2485 %
Root relative squared error             55.2243 %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.925     0.075     0.939      0.925     0.932      0.925    -1
                 0.925     0.075     0.907      0.925     0.916      0.925     1
Weighted Avg.    0.925     0.075     0.925      0.925     0.925      0.925

=== Confusion Matrix ===

    a    b   <-- classified as
 1373  112 |    a = -1
   89 1096 |    b = 1
```

Table 3: Summary of test results - Support Vector Machine

Summary	Error Rate	TP Rate	FP Rate	Precision	Recall	F-Measure
Full Training Set	7.603 %	0.924	0.077	0.924	0.924	0.924
Cross fold Validation	7.5281 %	0.925	0.075	0.925	0.925	0.925

Naïve Bayes

The naïve Bayes model calculates the number of actual and predicted class labels using a confusion matrix, and in turn computes the conditional probability in predicting the classes. The model built using weka with the count values corresponding to individual attributes is displayed below:

```
                          Class
Attribute                   -1       1
                         (0.56)  (0.44)
=====================================
Prefix_Suffix
  -1                      857.0    31.0
  1                       630.0  1156.0
  [total]                1487.0  1187.0

having_Sub_Domain
  -1                      495.0   110.0
  0                       481.0   383.0
  1                       512.0   695.0
  [total]                1488.0  1188.0

URL_of_Anchor
  -1                     1099.0    15.0
  1                       375.0  1170.0
  0                        14.0     3.0
  [total]                1488.0  1188.0

Request_URL
  1                       917.0   777.0
  -1                      249.0   328.0
  0                       322.0    83.0
  [total]                1488.0  1188.0

SSLfinal_State
  1                       185.0  1022.0
  -1                     1266.0   111.0
  0                        37.0    55.0
  [total]                1488.0  1188.0

Links_in_tags
  0                       548.0   451.0
  1                       601.0   638.0
  -1                      339.0    99.0
  [total]                1488.0  1188.0

SFH
  -1                     1211.0  1031.0
  1                       276.0   156.0
  [total]                1487.0  1187.0

age_of_domain
  -1                      603.0   183.0
  1                       884.0  1004.0
  [total]                1487.0  1187.0
```

Full Training Set

```
=== Evaluation on training set ===
=== Summary ===

Correctly Classified Instances        2503               93.7453 %
Incorrectly Classified Instances       167                6.2547 %
Kappa statistic                          0.874
K&B Relative Info Score             221230.8189 %
K&B Information Score                 2192.1489 bits      0.821  bits/instance
Class complexity | order 0            2645.6335 bits      0.9909 bits/instance
Class complexity | scheme              706.029  bits      0.2644 bits/instance
Complexity improvement     (Sf)       1939.6046 bits      0.7264 bits/instance
Mean absolute error                      0.0887
Root mean squared error                  0.227
Relative absolute error                 17.9706 %
Root relative squared error             45.6988 %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.923     0.045      0.963     0.923     0.943      0.979    -1
                 0.955     0.077      0.909     0.955     0.931      0.979    1
Weighted Avg.    0.937     0.059      0.939     0.937     0.938      0.979

=== Confusion Matrix ===

    a    b   <-- classified as
 1371  114 |    a = -1
   53 1132 |    b = 1
```

Cross Fold Validation

```
=== Stratified cross-validation ===
=== Summary ===

Correctly Classified Instances        2500               93.633  %
Incorrectly Classified Instances       170                6.367  %
Kappa statistic                          0.8717
K&B Relative Info Score             221008.7057 %
K&B Information Score                 2189.8016 bits      0.8202 bits/instance
Class complexity | order 0            2645.6399 bits      0.9909 bits/instance
Class complexity | scheme              711.4068 bits      0.2664 bits/instance
Complexity improvement     (Sf)       1934.2331 bits      0.7244 bits/instance
Mean absolute error                      0.0892
Root mean squared error                  0.2278
Relative absolute error                 18.0607 %
Root relative squared error             45.8488 %
Total Number of Instances             2670

=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.921     0.045      0.963     0.921     0.942      0.979    -1
                 0.955     0.079      0.906     0.955     0.93       0.979    1
Weighted Avg.    0.936     0.06       0.938     0.936     0.936      0.979

=== Confusion Matrix ===

    a    b   <-- classified as
 1368  117 |    a = -1
   53 1132 |    b = 1
```

Table 4: Summary of test results - Naive Bayes Model

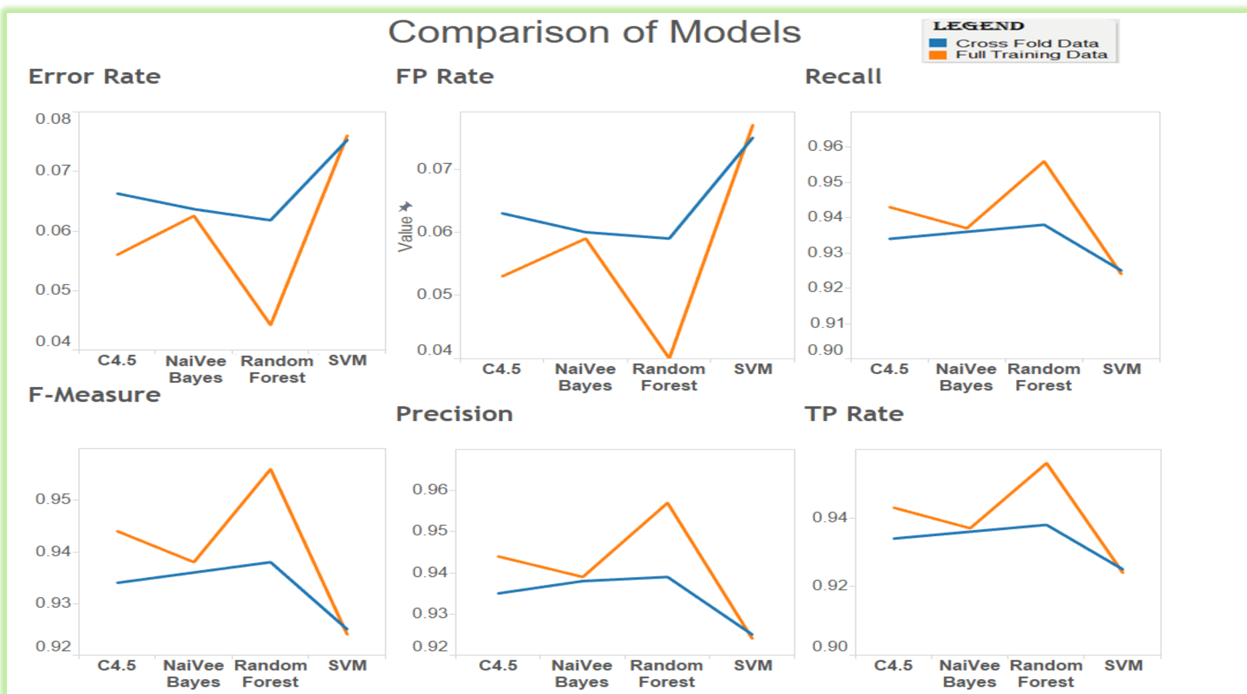
Summary	Error Rate	TP Rate	FP Rate	Precision	Recall	F-Measure
Full Training Set	6.2547 %	0.937	0.059	0.939	0.936	0.938
Cross fold Validation	6.367 %	0.936	0.060	0.938	0.936	0.936

FINAL MODEL SELECTION

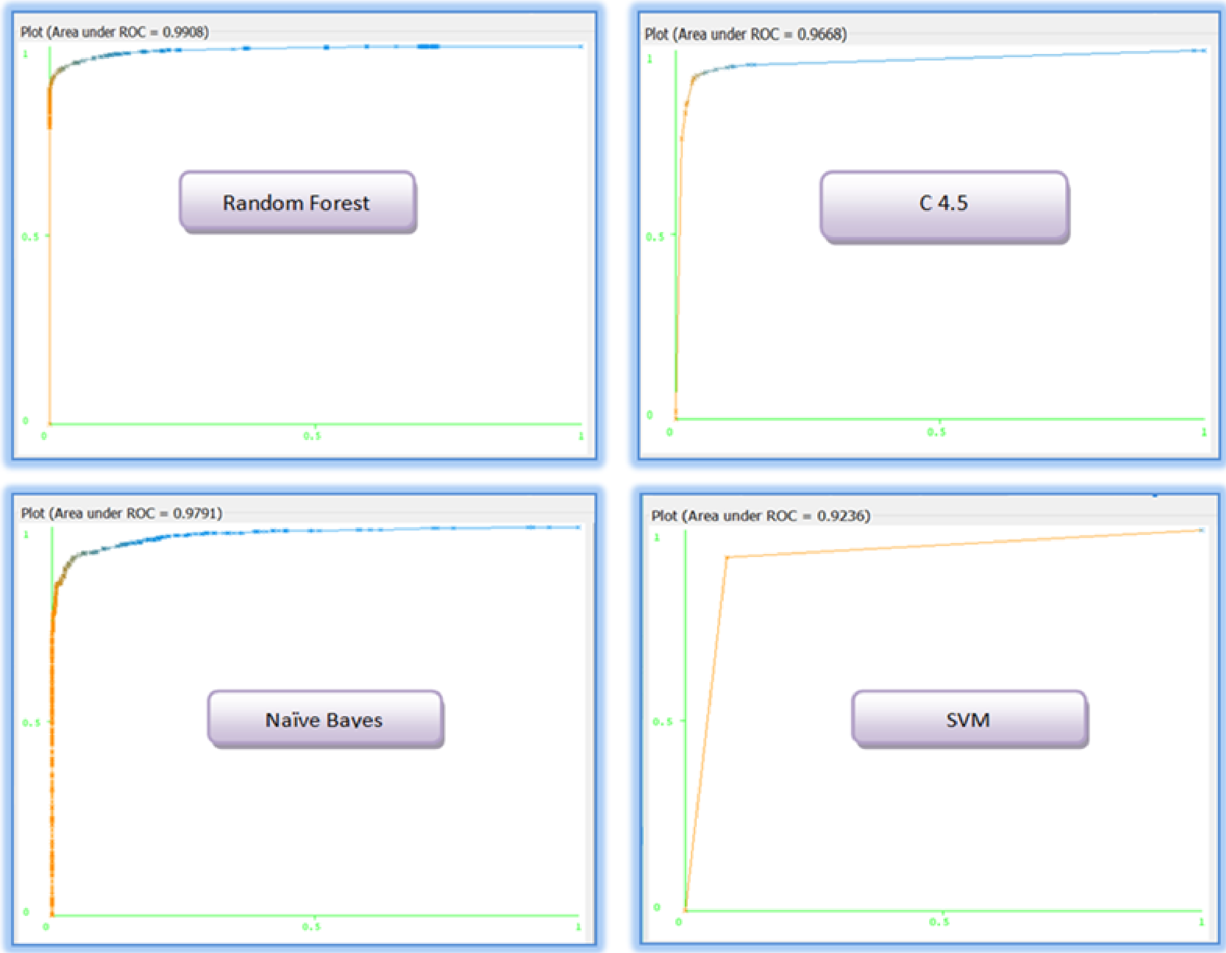
Below are the results of the all 4 models that were previous analyzed. The key parameters to check are Error Rate, False Positive, and F-measure.

Error Rate is the percentage of misclassified instances out of the total. False Positive are those that the model has classified as a phishing but are actually not phishing. F-measure is the harmonic mean of Recall and Precision, where maximum Precision indicates no false positives and maximum recall indicates no false negatives.

In all the below parameters, Random Forest turns out to be the best model when tested on both the full training data and the cross-fold data.



Below are the Receiver_operating_characteristic or ROC curve and plots the true positive rate against the false positive rate. The closer a curve is to the point (1,0) the better the model is able to correctly classify the data. We see below that Random Forest quite outperforms the other algorithms

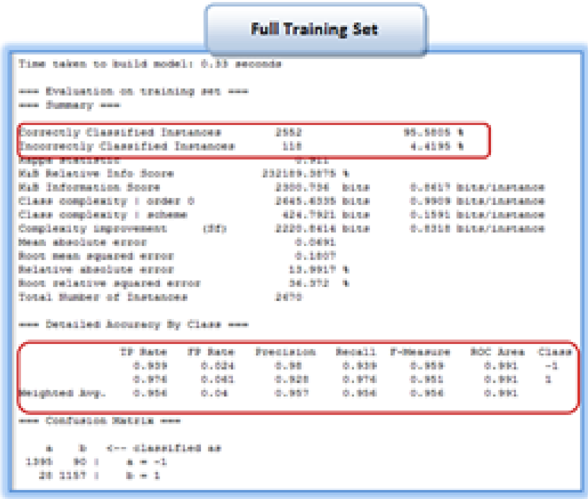


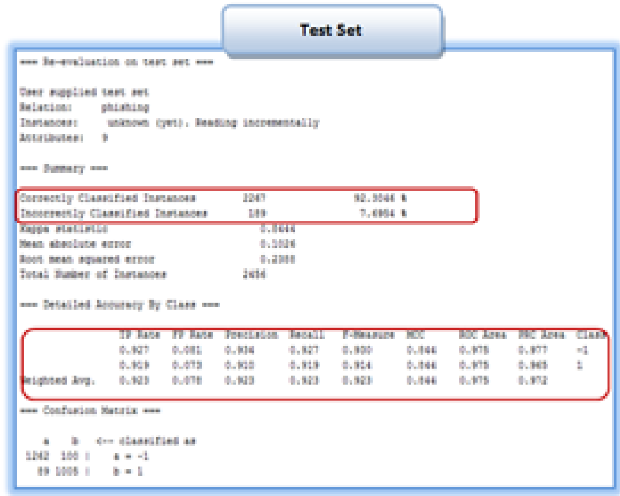
The area under the curve below goes further to show that Random Forest is the best classifier.

Model	Area under ROC
Random Forest	0.9908
Naïve Bayes	0.9668
C 4.5	0.9791
SVM	0.9236

Since we have concluded that Random Forest is the best performing model on the training ad cross-fold sets, we now go ahead and validate it on the testing set.

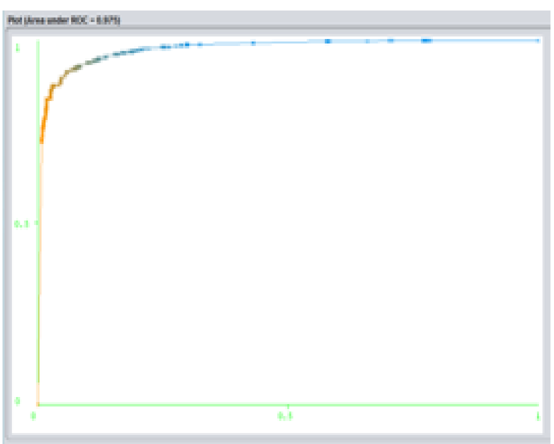
From the results below we see that the model performs very well on the testing set too. We expect that the model's performance will dip when applied on the testing set and here too we see that all the parameters are affected, the Error rate has increased, False Positives have increased etc. But this is an expected result and the change is in an acceptable range.





Summary	Error Rate	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area
Full Training Set	4.4195 %	0.956	0.040	0.957	0.956	0.956	0.990
Test Set Validation	7.6954 %	0.923	0.078	0.923	0.923	0.923	0.972

Finally, looking at the ROC, we see that it is not as good in classifying the testing set, however this too is in an acceptable range and an expected result.

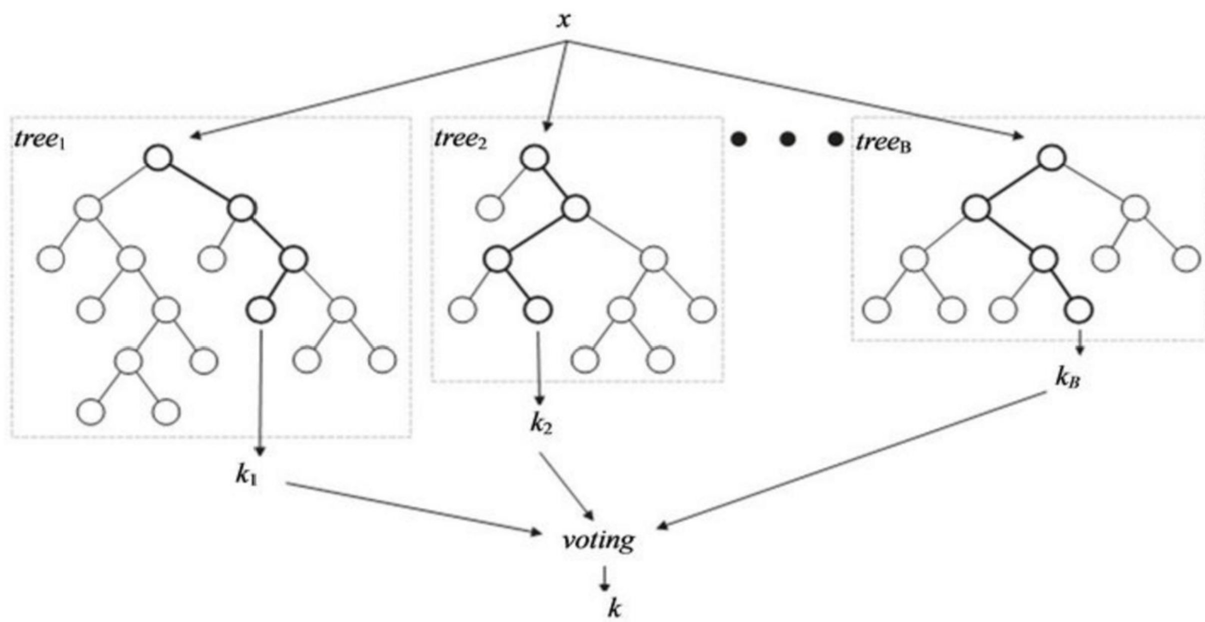


CONCLUSION

To summarize, in this project we have selected the best URL and Content-Based attributes for classifying Phishing Webpages. Further, we have demonstrated that the proposed features are highly relevant to the automatic discovery and classification of phishing websites.

We evaluated four algorithms – C4.5, Random Forest, SVM and Naives Bayes and studied their benefits and trade offs. Random Forest outperformed the other models in all parameters and also validated well on the testing dataset.

We believe that the Random Forest while computationally expensive and constantly changes is the best fit for our data as there is constant changes in the phishing website themselves. Random Forest as shown in the below diagram creates many trees and employs a voting system to create one final best fit tree. This was it is able to analyze all the attributes and select the best ones for classification.



This area of study is constantly changing and is getting more difficult to catch as hackers constantly challenge existing systems in the effort steal data, identities and ultimately money. We believe our study will help analysts in the area of URL and Content-based selection and model building.

REFERENCES

Data Source : http://eprints.hud.ac.uk/24330/

http://isyou.info/jisis/vol4/no3/jisis-2014-vol4-no3-02.pdf

http://docs.apwg.org/reports/apwg_trends_report_q4_2015.pdf

http://ijarcet.org/wp-content/uploads/IJARCET-VOL-3-ISSUE-5-1584-1589.pdf

www.phishing.org

http://weka.sourceforge.net/doc.dev/weka/attributeSelection/BestFirst.html