Phishing Web Sites Features Classification Based on Extreme Learning Machine

Nimisha S
MTech Student, Department of Artificial Intelligence and Engineering, CMR University, Bengaluru, IndiaEmail: nimisha.sivanesan@gmail.com

Dr S Saravana Kumar
Professor and Head of Computer Science and Engineering PG department, CMR University, Bengaluru, India Email: saravanakumarmithun@gmail.com

Dr.Rubini P
Professor and Head of Computer Science and Engineering, CMR University, Bengaluru, IndiaEmail: rubini.p@cmr.edu.in India

Abstract:
Some of the highly frequent as well as hazardous computer crimes assaults is phishing. The archival documents by people and companies to execute transactions are the target of these assaults. Phishing websites use a variety of indicators in both their text and data that is dependent on internet browsers. Phishing is an unique kind of networking attack in which the perpetrator makes a copy of a current Web page to trick users into providing private, economic, perhaps credential information towards what they believe is actual service provider's Web site (e.g., through employing carefully formulated e-mails or instant chats). The objective of this project is to classify 30 variables, comprising data from phishing websites, using Extreme Learning Machines (ELM) on a database at UC Irvine.

Index Terms–Random forest classifier, Adaboost classifier, Support vector classifier, Phishing.

I. INTRODUCTION

Internet use has become an essential part of our daily activities as a result of rapidly growing technology. Due to this rapid growth of technology and intensive use of digital systems, data security of these systems has gained great importance. The primary objective of maintaining security in information technologies is to ensure that necessary precautions are taken against threats and dangers likely to be faced by users during the use of these technologies.

The word „Phishing" initially emerged in 1990s. The early hackers often use „ph" to replace „f" to produce new words in the hacker's community, since they usually hack by phones. Phishing is a new word produced from „fishing", it refers to the act that the attacker allure users to visit a faked Website by sending them faked e-mails (or instant messages), and stealthily get victim's personal information such as user name, password, and national security ID, etc. This information then can be used for future target advertisements or even identity theft attacks (e.g., transfer money from victims" bank account). The frequently used attack method is to send e-mails to potential victims, which seemed to be sent by banks, online organizations, or ISPs. In these e-mails, they will make up some causes, e.g. the password of your credit card had been mis-entered for many times, or they are providing upgrading services, to allure you visit their Web site to conform or modify your account number and password through the hyperlink provided in the e-mail. You will then be linked to a counterfeited Web site after clicking those links. The style, the functions performed, sometimes even the URL of these faked Web sites are similar to the real Web site.

It"s very difficult for you to know that you are actually visiting a malicious site. If you input the account number and password, the attackers then successfully collect the information at the server side, and is able to perform their next step actions with that information (e.g., withdraw money out from your account). Phishing itself is not a new concept, but it's increasingly used by phishers to steal user information and perform business crime in recent years. Within one to two years, the number of phishing attacks increased dramatically. According to Gartner Inc., for the 12 months ending April 2004, there were1.8 million phishing attack victims, and the fraud incurred by phishing victims totalled $1.2 billion.

II. LITERATURE SURVEY

The paper by Rami M Mohammad [1] et al they shed light on the important features that distinguish phishing websites from legitimate ones and assess how rule-based classification data mining techniques are applicable in predicting phishing websites. We also experimentally show the ideal rule-based classification technique for detecting phishing.

According to Rami M Mohammad [2] et al here they explored important features that are automatically extracted from websites using a new tool instead of relying on an experienced human in the extraction process and then judge on the features importance in deciding website legitimacy. Our research aims to develop a group of features that have been shown to be sound and effective in predicting phishing websites and to extract those features according to new scientific precise rules.

Here Neda Abdelhamid [3] deals with the problem by proposing an AC algorithm called Enhanced Multi-label Classifiers based Associative Classification (eMCAC). This algorithm discovers rules associated with a set of classes from single label data that other current AC algorithms are unable to induce.

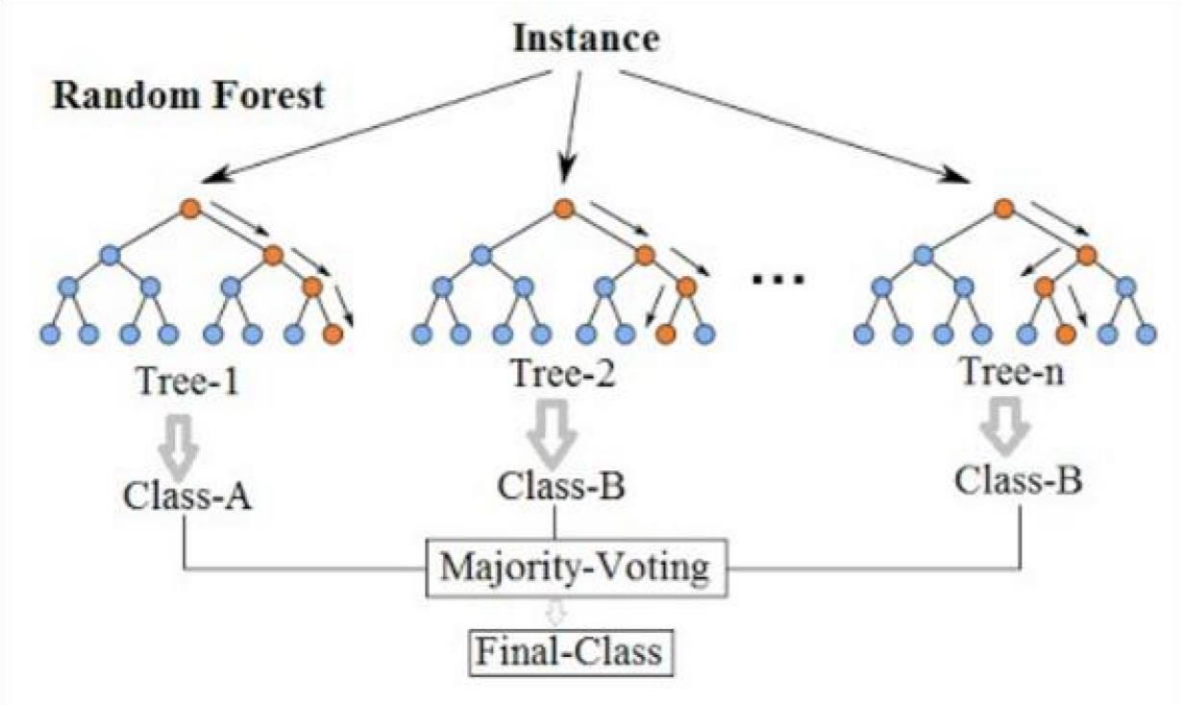
By Weider D. Yu [4] et al explains in detail the various methods used in phishing. Here they perform a root-cause analysis of the methods used in phishing, the motivation for phishing and in the process come up with a fishbone diagram outlining the causes and methodologies used in phishing. This analysis is aimed at helping developers to design and develop better anti phishing solutions.

Ying Pan and Xuhua Ding [5] proposed a novel approach, which is independent of any specific phishing implementation. Our idea is to examine the anomalies in web pages, in particular, the discrepancy between a web site‟s identity and its structural features and HTTP transactions. It demands neither user expertise nor prior knowledge of the website.

III. PYTHON IMPLEMENTATION

Random Forest Algorithm:

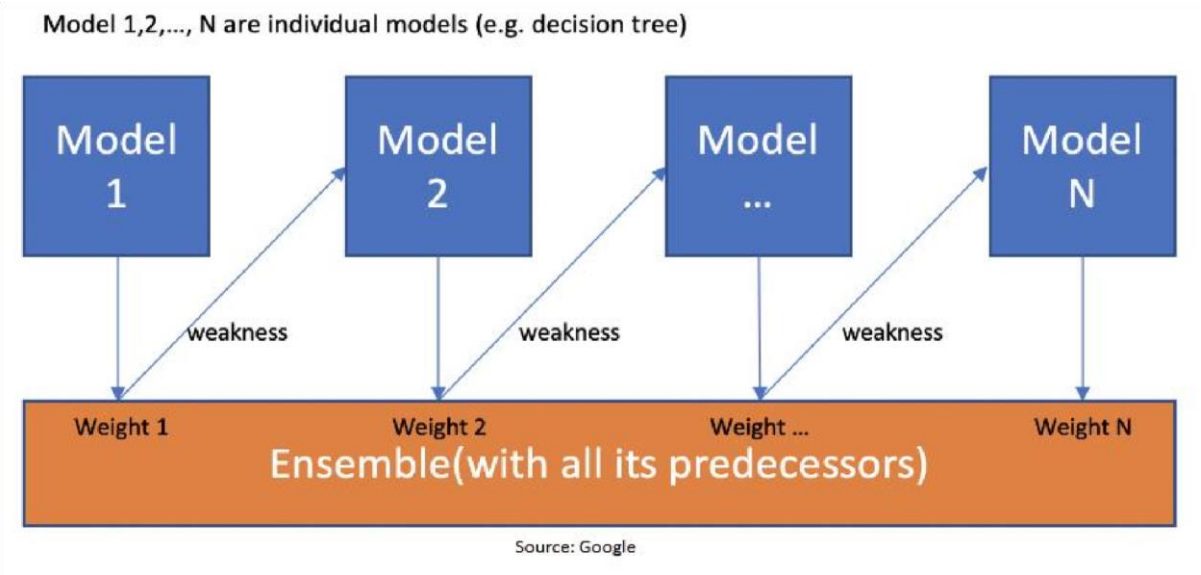
Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of over fitting to their training set.



AdaBoost Classifier:

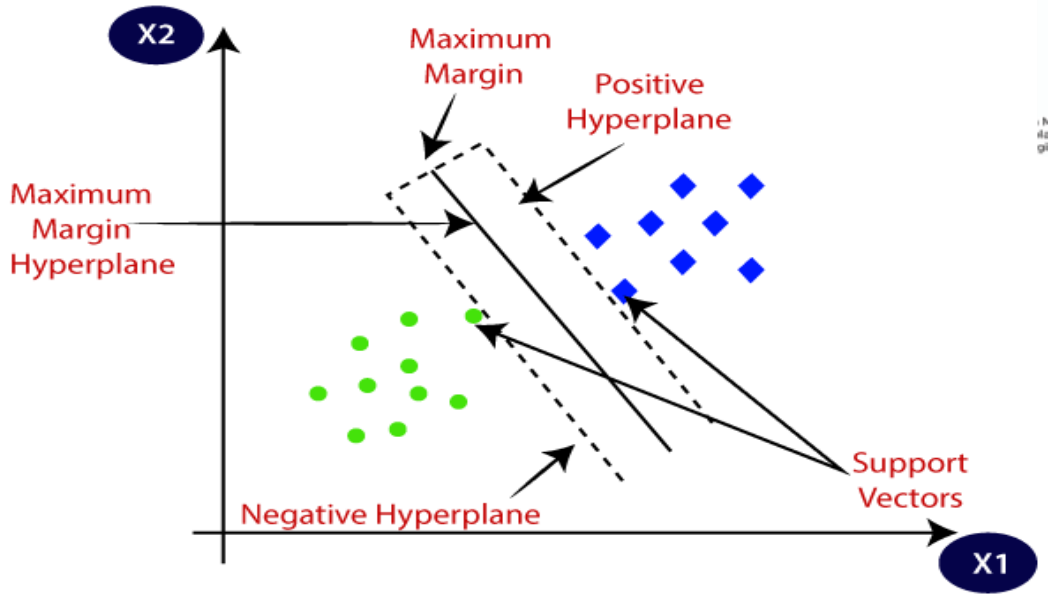
AdaBoost algorithm, short for Adaptive Boosting, is a Boosting technique used as an Ensemble Method in Machine Learning. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances. Boosting is used to reduce bias as well as variance for supervised learning.

It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is grown from previously grown learners. In simple words, weak learners are converted into strong ones. The AdaBoost algorithm works on the same principle as boosting with a slight difference.



Support Vector Classifier:

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.SVM chooses the extreme points/vectors that help in creating the hyper plane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyper plane:



IV. PROPOSED SYSTEM

Here we defined various features of phishing attack and we proposed a classification model in order to classification of the phishing attacks. This method consists of feature extraction from websites and classification section. In the feature extraction, we have clearly defined rules of phishing feature extraction and these rules have been used for obtaining features.

In this paper, our proposed system provides methods to develop machine learning phishing attack classifier based on Feed-Forward Neural Networks, with intent of improving the accuracy.

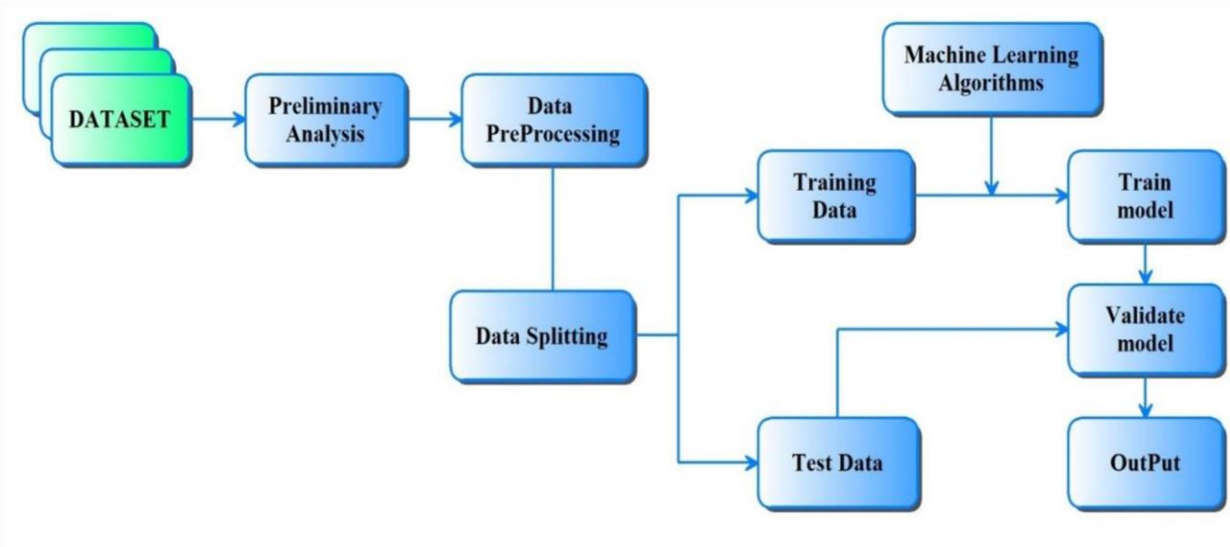


Fig-1 System architecture

V. GRAPHS

The pandas library is typically used to read and analyse datasets that are both numerical and textual. After then, the read head function was used to show the data that had been successfully read.

```
data.head()
```

	Index	UsingIP	LongURL	ShortURL	Symbol@	Redirecting//	PrefixSuffix-	SubDomains	HTTPS	DomainRegLen	Favicon	I
0	0	1	1	1	1	1	-1	0	1	-1	1	
1	1	1	0	1	1	1	-1	-1	-1	-1	1	
2	2	1	0	1	1	1	-1	-1	-1	1	1	
3	3	1	0	-1	1	1	-1	1	1	-1	1	
4	4	-1	0	-1	1	-1	-1	1	1	-1	1	

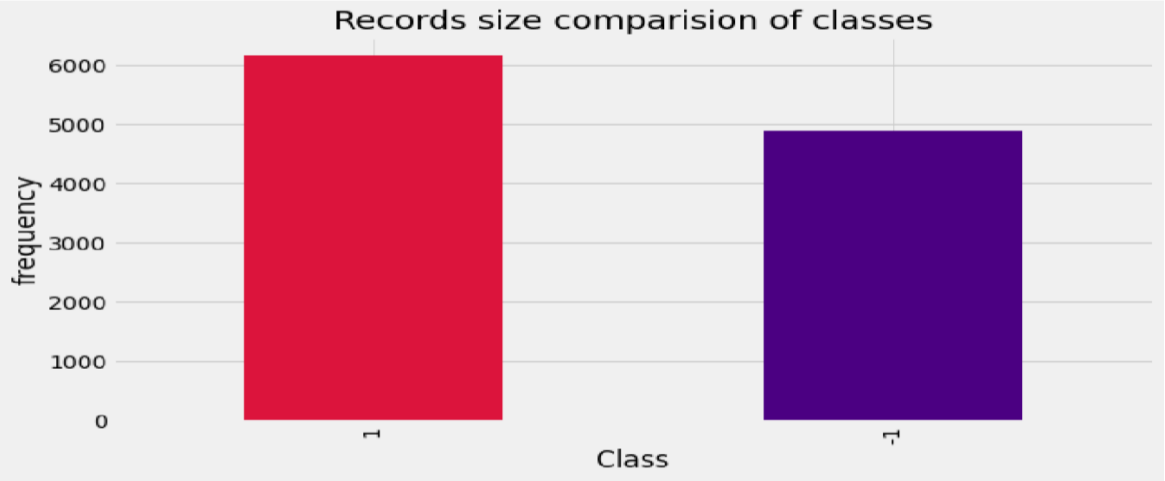


Fig-2 Records size comparison of classes

Two classes are participating in this web phishing experiment. One of them is web phishing, while the other is not. We compared the size of the records for each class in the graph above using a bar chart.

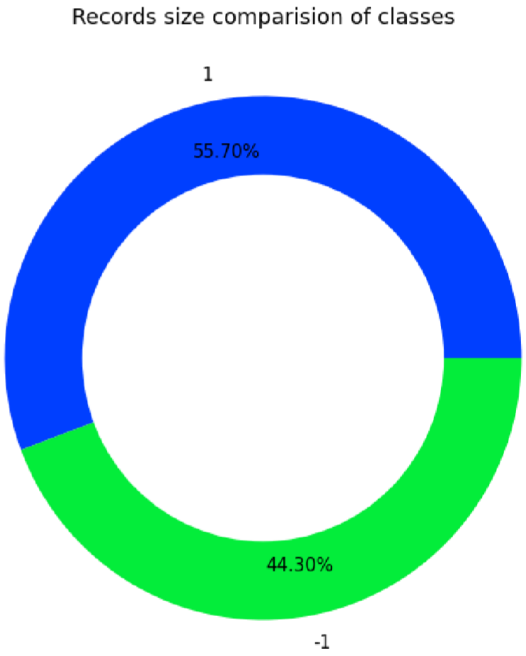


Fig-3 Record size comparison of classes

An example of a pie chart is the graph above. It displays the outcome of the records size comparison for 100%. We are able to determine the data's balance or imbalance using this graph.

VI. RESULT ANALYSIS

i) Classification Report of Random forest classifier

A Classification report is used to measure the quality of predictions from a classification algorithm. The report shows the main classification metrics precision, recall and f1-score on a per-class basis. The metrics are calculated by using true and false positives, true and false negatives.

```
print(classification_report(y_true=y_test, y_pred= RF_predictions,target_names=["No","Yes"]))
```

	precision	recall	f1-score	support
No	0.97	0.96	0.96	976
Yes	0.97	0.98	0.97	1235
accuracy			0.97	2211
macro avg	0.97	0.97	0.97	2211
weighted avg	0.97	0.97	0.97	2211

j) Confusion matrix of Random forest classifier

A confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. When we get the data, after data cleaning, pre-processing, and wrangling, the first step we do is to feed it to an outstanding model and of course, get output in probabilities. But hold on! How in the hell can we measure the effectiveness of our model. Better the effectiveness, better the performance, and that is exactly what we want. And it is where the Confusion matrix comes into the limelight. Confusion Matrix is a performance measurement for machine learning classification.

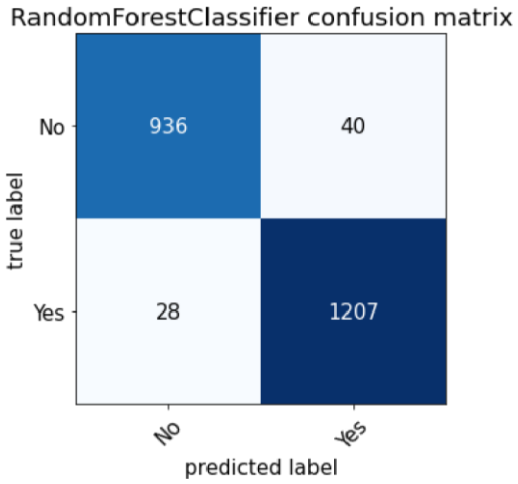


Fig-4 confusion matrix of RFM

k) Classification Report of Adaboost classifier

A Classification report is used to measure the quality of predictions from a classification algorithm. How many predictions are True and how many are False. More specifically, True Positives, False Positives, True negatives and False Negatives are used to predict the metrics of a classification report.

```
print(classification_report(y_true=y_test, y_pred= ABC_predictions,target_names=["No","Yes"]))
```

	precision	recall	f1-score	support
No	0.94	0.92	0.93	976
Yes	0.94	0.96	0.95	1235
accuracy			0.94	2211
macro avg	0.94	0.94	0.94	2211
weighted avg	0.94	0.94	0.94	2211

Adaboost classifier model's classification accuracy rate is 0.98%

l) Confusion matrix of Adaboost classifier

Well, it is a performance measurement for machine learning classification problem where output can be two or more classes. It is a table with 4 different combinations of predicted and actual values.

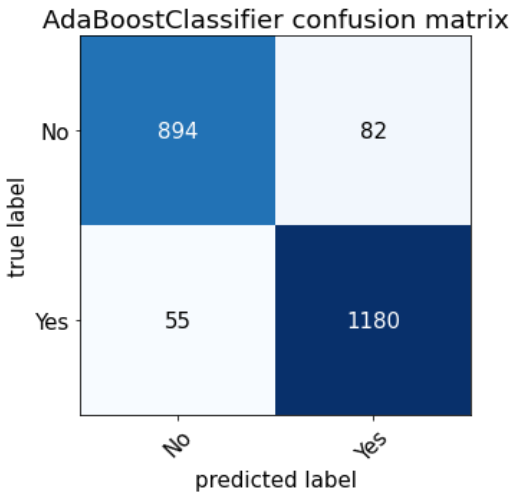


Fig-5 Confusion matrix of ABC

From the Class "NO" 894 records correctly predicted as class "NO" and 82 records incorrectly predicted as class "YES". From the class "YES" 55 records incorrectly predicted as class "NO" and 1180 records correctly predicted as class "YES".

m) Classification Report of Support vector classifier

Support vector classifier model's classification accuracy is 0.95%

```
print(classification_report(y_true=y_test, y_pred= svc_predictions,target_names=["No","Yes"]))
```

	precision	recall	f1-score	support
No	0.96	0.92	0.94	976
Yes	0.94	0.97	0.96	1235
accuracy			0.95	2211
macro avg	0.95	0.95	0.95	2211
weighted avg	0.95	0.95	0.95	2211

n) Confusion matrix of Support vector classifier

From the Class "NO" 901 records correctly predicted as class "NO" and 75 records incorrectly predicted as class "YES". From the class "YES" 33 records incorrectly predicted as class "NO" and 1202 records correctly predicted as class "YES".

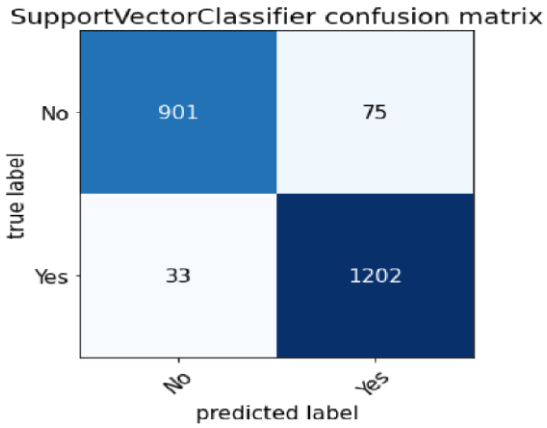


Fig-6 Confusion matrix of SVC

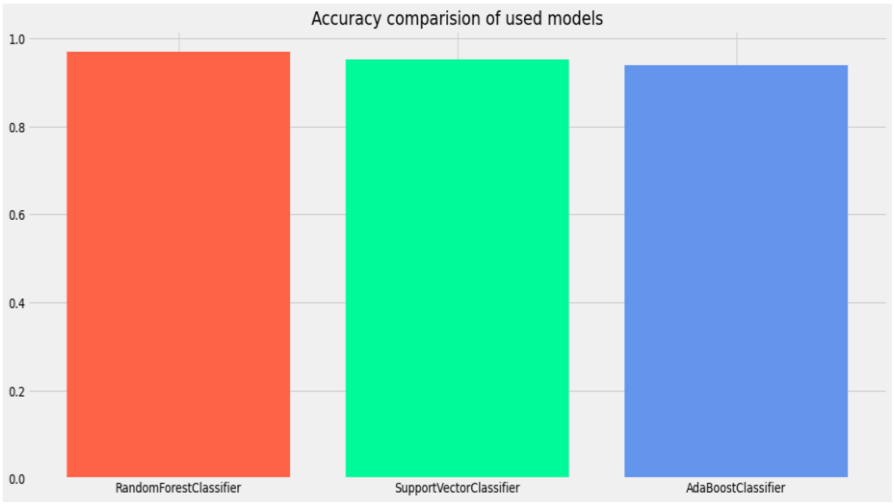


Fig-7 Accuracy comparison of used models

The model performance outcome is represented graphically in this chart. This bar-chart was made with the help of the Matplotlib library. We may use labels to determine the provided model's optimal performance based on the height of the bar.

VII. CONCLUSION

In this paper, we defined features of phishing attack and we proposed a classification model in order to classification of the phishing attacks. This method consists of feature extraction from websites and classification section. In the feature extraction, we have clearly defined rules of phishing feature extraction and these rules have been used for obtaining features. In order to classification of these features, SVC, RFC and ABC were used. In different activation functions were used and achieved highest accuracy score.

VIII. REFERENCES

[1] L. McCluskey, F. Thabtah, and R. M. Mohammad. “Intelligent rule based phishing websites classification” (2014). IET Inf. Secure. 8(3):153–160.

[2] R. M. Mohammad, F. Thabtah, and L. McCluskey. “Predicting phishing websites based on self-structuring neural network” (2014). Neural Comput.Appl.25(2):443–458.

[3] N. Abdelhamid. “Multi-label rules for phishing classification” (2015). Appl. Comput. Informatics.11 (1): 29–46.

[4] W. D. Yu, S. Nargundkar, and N. Tiruthani. “A phishing vulnerability analysis of web-based systems” (2008). IEEE Symp. Comput. Commun. (ISCC).326–331.

[5] P. Ying and D. Xuhua. “Anomaly based web phishing page detection” (2006) in Proceedings - Annual Computer Security Applications Conference, ACSAC.381–390.