Michigan Technological University

CS 5472 – ADVANCED TOPICS IN COMPUTER SECURITY

Term Project – Spring 2018

ENHANCEMENT IN PHISHING WEBSITE DETECTION USING RANDOM FOREST CLASSIFIER

AUTHOR

KARAN SUNCHANAKOTA

SUPERVISOR

DR. BO CHEN

Table of Contents

I. INTRODUCTION

Phishing is a widely known act of the attackers stealing the confidential information (sometimes, money) of the users by spoofing the websites or by luring the users to visit some fake sites where they disclose their personal information open to the attackers, though done unintentionally and innocently. According to criminal laws, this act is a deliberate deception made for the sole aim of personal gains or for smearing an individual's image. Usually, attackers create a replica website of a legitimate organization and using that website they attempt to electronically obtain delicate or confidential information from users. Phishing is usually committed with the aid of an electronic device (such as Tablets and computer) and a computer network. Phishing attackers usually perform their evil by communicating well-composed messages (known as social engineered messages) to users in order to persuade them to reveal their personal information which will be used by the fraudster to gain unauthorized access to the user's account. [1]

For example, a phishing email sent to a user might contain a malware (called man in the browser (MITB)), this malware could be in the form of web browser ActiveX components, plugins, or email attachments; if this user ignorantly download this attachment to his pc, the malware will install itself on the user's pc and would in turn transfer money to the fraudster's bank account whenever the user (i.e., the legitimate owner of the bank account) tries to perform an online transaction.

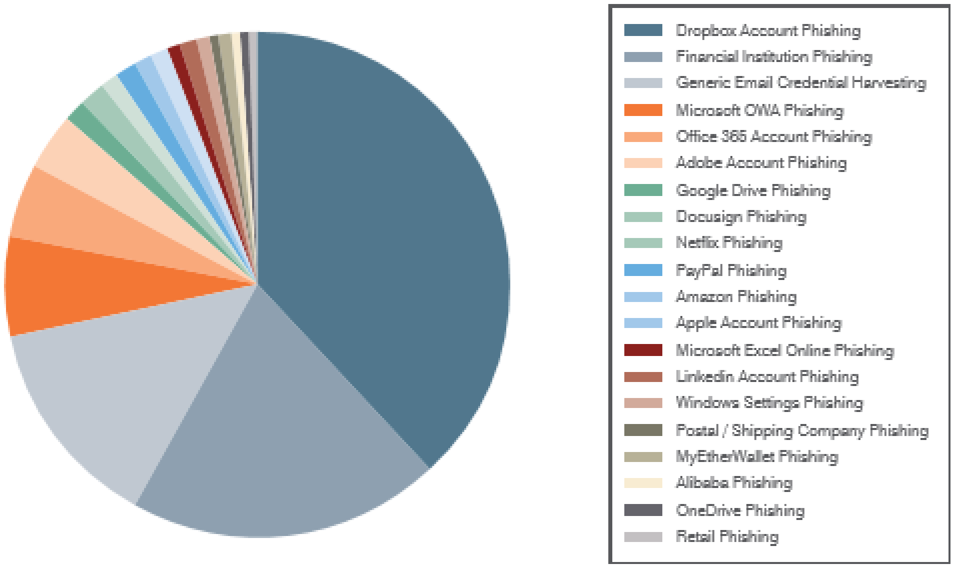
Types of Phishing:

- **Spear Phishing:** Phishing attack that aims to attack a specific organization or a person is named as spear phishing. Attackers may accumulate individual data about their target to expand their likelihood of accomplishment. This method is by a wide margin the best on the web today, representing 91% of a security breach.
- **Clone Phishing:** Phishing attack whereby a genuine, and previously sent, an email containing a connection or connection has had its substance and beneficiary address taken and used to make a relatively indistinguishable or cloned email is known as Clone Phishing. The connection or connection inside the email is supplanted with a malevolent form and after that sent from an email deliver ridiculed to seem to originate from the first sender.
- **Whaling:** In whaling, the disguising site page/email will take a more genuine official level shape. The substance will be created to focus on an upper administrator and the individual's part in the organization. The substance of a whaling attack email is regularly composed as a legitimate subpoena, client grievance, or official issue. Whaling trick messages are intended to take on the appearance of a basic business email, sent from a genuine business expert.
- **Link Manipulation:** In this, phishers utilize some type of specialized duplicity intended to make a connection in an email and they claim that they are a legitimate organization. They can be easily identified by spelling errors in URL etc.

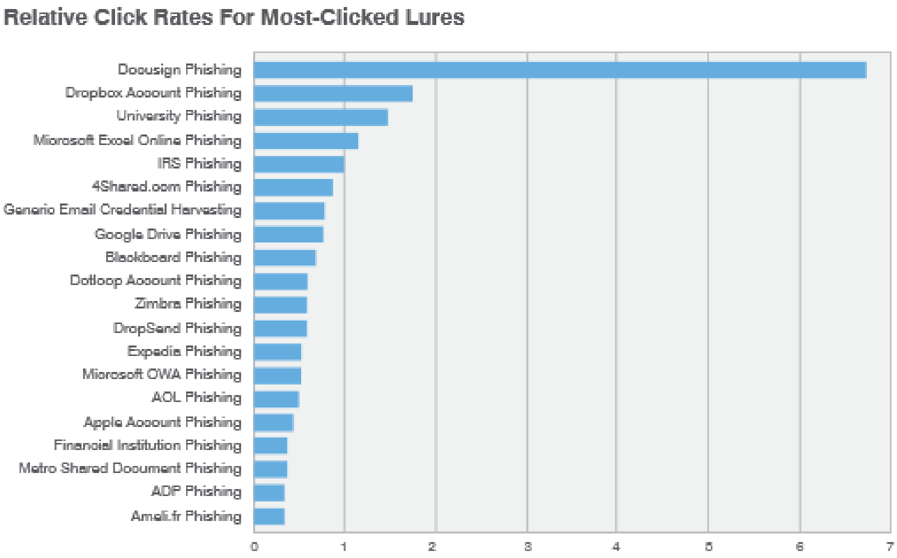
- **Website Forgery:** In this, phishers utilize JavaScript commands keeping in mind the end goal to modify the address bar. This is done either by placing a photo of a benign URL over the address bar or by closing the first bar and opening up another one with the benign URL.
- **Covert redirect:** It is an inconspicuous strategy to perform phishing attacks that makes links to seem benign, yet really divert a user to a phishing site. This is generally disguised under a sign in pop up in view of an influenced site's domain.
- **SMS and Voice Phishing:** Invoice phishing, attackers utilize counterfeit guest ID information to give the appearance that calls originate from a trusted organization. In SMS phishing, utilizes mobile phone instant messages to lure users to obtain sensitive data.

In 2018, phishing lures associated with Dropbox file-sharing far outnumbered any other lure, accounting for a disproportionately high volume of phishing activity (more than a third of all tracked lures). The next most prevalent lures were those related to financial institutions and generic email credential harvesting.

Relative message volume for the top 20 phishing lures in 2018



Interestingly, although there were far more Dropbox lures in play, it was DocuSign lures that garnered the highest relative click rate (nearly 7 percent). DocuSign lures were approximately 3x more effective than Dropbox lures. In fact, DocuSign click rates beat out all other credential phishing email lures by a long shot. [2]



Average click rates for the top 10 lures

II. BACKGROUND AND RELATED WORK

To achieve an honest and phishing websites detection, two different approaches are used. The first one checks if the requested URL is on the blacklists by comparing with those in that list. Meta-heuristic methods in which quite a lot of features are collected from the website to categorize it as either legitimate or phishing website is the second approach. The accurateness of the meta-heuristic method is based on extracting a set of distinguishing features which may help in differentiating the website. Data mining or Machine Learning techniques are generally used to extract the features from the websites to find patterns as well as relationships between them. Machine Learning algorithms are highly imperative for decision-making since decisions can be made based on the rules accomplished from a data-mining algorithm. [3]

Hugh Corporate Organizations adopt a content-based approach for phishing website detection. In this approach, the attacks are identified by analyzing the content of the site. Features utilized as a part of this approach incorporate spelling mistakes, source of the images, links, password fields, embedded links, etc. alongside URL and host-based features. For example, Google has its own anti-phishing filter like SpoofGuard, CANTINA, etc. that detects phishing and malware information by checking the content of URL like HTML tags, JavaScript elements. The classifier is regularly re-trained according to new trends in phishing. Few researchers used fuzzy logic and fingerprinting approaches to detect phishing sites.

An intelligent system for phishing webpage detection in e-banking is proposed by Aburrous et al. [4]. They proposed a model based on fuzzy logic combined with data mining algorithms to examine the techniques by describing the phishing website aspects and by categorizing the phishing types. By using 10-fold cross-validation, they achieved 86.38% classification accuracy, which is very low.

He et al. [5], proposed a model based on HTTP transaction, page content, and search engine results, they detected phishing pages with 97% of classification accuracy. A new type of intelligent algorithm based on approximate string matching is used by Arade et al. [6] to compare the addresses in the database of the proposed system and the webpage address. In this study, the problem is with the probability of occurring false positive occurrence, means legitimate webpages can be considered as phishing webpages. A model for detecting phishing webpages is proposed by Shahriar & Zulkernine [7] using the reliability of suspected pages. In their study, a finite state machine is proposed to assess webpage behavior by tracing the webpage form the submission as well as from the corresponding responses. MCAR is presented as a phishing detection method by Ajlouni et al. [8] by adopting the features from Aburrous et al. work by achieving 98.5% accuracy in classifying the webpages, but they did not give any information about how many rules were extracted by using the MCAR algorithm. A rule-based model in which Neuro-Fuzzy classifier with five inputs employed to detect phishing websites was proposed by Barraclough et al. [9]. The proposed model accuracy was 98.5%. Another approach, which uses the webpage under scrutiny and distinguishes all the direct and indirect links related to the page, was proposed by Ramesh et al. [10]. The indirect page links are taken out from the search engine result, but the direct links are taken out from the page content itself. In order to map the domains of the suspicious webpage and phishing target related to IP, third-party DNS lookup is also used. They achieved 99.62% accuracy to detect phishing webpages, but the method has external dependency which is 3rd-party DNS lookup and search engine result. Moreover, phishing webpages hosted on the compromised domains cannot be detected. Another detection model with a set of conventional features is proposed by Mohammed et al. [11] and calculated the detection error-rate yielded by the set of associative classification algorithms. The results presented that C4.5 has an average error rate of 5.76%. In order to extract the rules from training data, Abdelhamid et al. [12] proposed a Multi-label Classifier based Associative Classification (MCAC). The limitation of the proposed model is that the induction of rules needs a large number of rules. They achieved a 97.5% classification accuracy. Zhang et al. [13] used Sequential Minimal Optimization classifier with five features to distinguish Chinese phishing websites. The limitation of this approach is that the extracted features are only for the detection of phishing webpages with the Chinese language [14]. Li et. al. [15] used the transudative support vector machine to detect and classify phishing web pages. They extract the features of the web page image to reflect the characteristics of web pages absolutely. Montazer et. al. [16] used fuzzy logic combined with rough sets-based data mining algorithm for phishing detection. A method based on the differences between the phishing websites and the imitated target websites was proposed Li et. al. [17] and they used the ball-based SVM algorithm to distinguish phishing website. Moghimi et al. [14] used approximate string-matching algorithms with all individual page resource elements and page hyperlinks instead of comparing them directly.

III. APPROACH

The existing approach uses a pattern recognition algorithm for phishing website detection. Following are the feature extraction that has been used in this approach.

1. **IP address:** Phishers use IP address in URL to steal information from users such as “http://125.98.3.123/fake.html. If IP address present, then it is phishing site.
2. **Tiny URL:** Attackers shorten length of URL and this URL directs into other web page which obtains data of user, such as such as such as "bit.ly/19DXSk4". If URL is shortened, then phishing.
3. **Long URL:** Phishers use lengthy URL to hide the malicious part. If length of URL greater than 80 it is phishing.
4. **"@" symbol:** 11 Adversaries use "@" symbol in URL, which redirects site to another website owned by attacker. If the symbol is present it is phishing.
5. **"//" symbol:** The "//" symbol is present only once in URL i.e after the protocol such as https:// , but if it present more than once it means it will be redirected to another website. So, two occurrences of "//" symbol, means phishing site.
6. **"-" symbol:** To make phishing site look benign, phishers use this symbol in URL. If it is present then it is phishing site.
7. **Dots in subdomain:** To look as genuine site, phishers place more number of dots in subdomain which redirects to different website ,such as "http://updateyouraccount.now.pp2.clickcom.com/cm " If dots in subdomain greater than 2 ,then it is phishing site.
8. **Domain expiry age:** Most of phishing domains live for very short span. The legitimate sites pay for the domain in advance so as to extend the life span. If domain expiry age is less than an year, then it is a phishing site.
9. **Favicon:** A favicon is a graphic icon associated with a specific web-page. Phishers use this so as to lure the users and make them believe that it is a benign site. If this image is loaded from different URL other than the one in address bar , then it is phishing.
10. **Special Characters:** Special characters like , _ etc. are used in the domain to lure the user as if they are legitimate URL. If these characters are present then it is phishing site.
11. **Abnormal URL:** This is feature where host name is not include in URL. If it is not present then it is phishing site.
12. **Age of domain:** Most of the phishing domains are created only, so the creation age of domain is less. Minimum age for genuine domain is 6 months. If creation age is less than 6 months, then it is phishing.
13. **Website Rank:** In this, the popularity of the website is determined by the number of visitors and the number of pages visited. As phishing websites live for a short period of time, they are recognized by the Alexa database. So, if rank is greater than 100,000 or there is no rank then it is phishing.

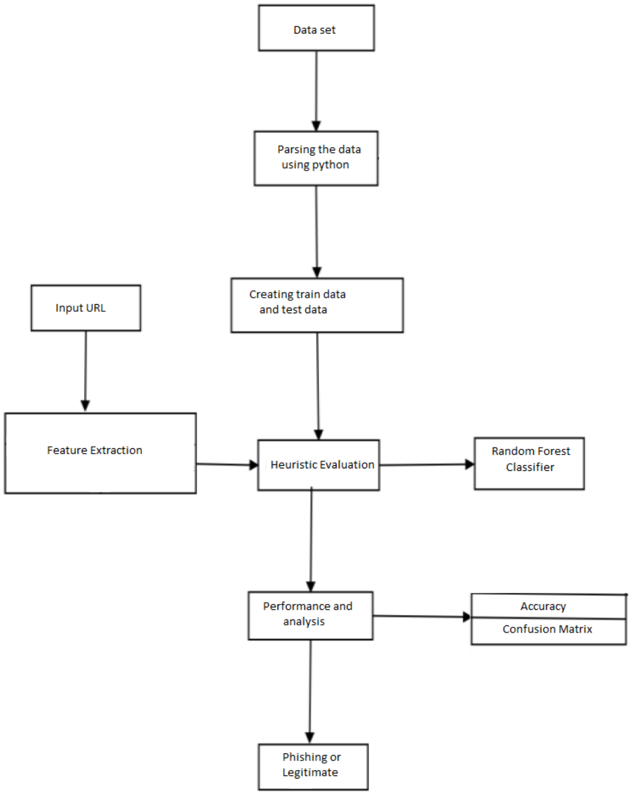
LIMITATIONS:

- Though the existing approach gives the output of whether the website is phishing or not. It is not completely reliable to believe the features.
- There is a huge possibility that features may result in a legitimate website to be a malicious website.
- For example, if a legitimate site like ‘google.com’ has its expiration date as today and didn’t renew it, as per the expiry age feature in the algorithm it returns ‘google.com’ as phishing.
- Another example is web traffic, as per the algorithm if the rank of the website is more than 100000, it is considered to be malicious. Consider if a good website has its rank of 100001 it returns good site as legitimate.
- The probability of imposing the legitimate site to be malicious site is high in this scenario.
- This leads to inconsistency of the algorithm.

CURRENT APPROACH

The current approach adds a random forest classifier predictor taken from a dataset of legitimate and phishing sites. The dataset needed for the entire procedure is gathered from Phish tank and since there was a large amount of data to process, parsing was performed primarily. Parsing is done to analyze the feature set. As per the above approach, we have 13 features that are by parsing and by rigorous analysis. Using pandas library from python parsing data sets into data frames i.e. CSV files can be created with the required feature data as columns.

FLOW CHART



The parsed dataset undergoes heuristic classification where the dataset is split into 70% and 30%. The 70% data is considered for training and 30% for testing. Using the libraries of random forest and inbuilt python functions, the classification model is constructed, and this model is tested using testing data. Using this model, other URLs of different websites that are input by the user are predicted. The last phase in the model to be performed is Performance Analysis which was done using confusion matrix and accuracy.

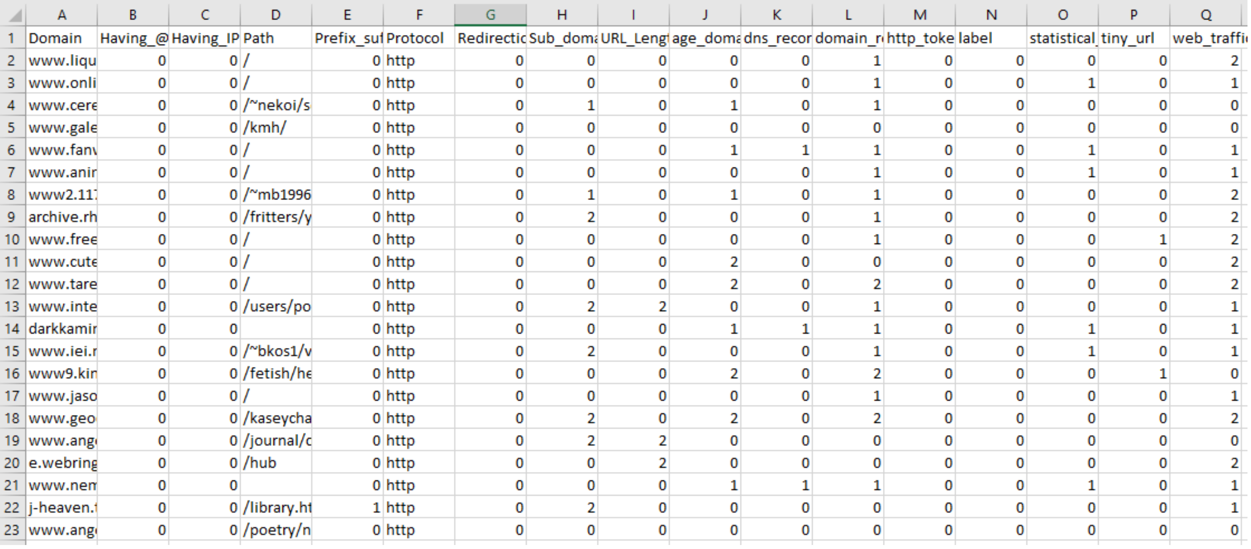
IV. EVALUATION

The evaluation consists of two parts. In the first part, the data set of legitimate set and phishing set is dividing into training and testing data where train data has 70% and test data has 30%. In the second part, accuracy and confusion matrix are created, using this data feature importance will be created through which a URL prediction can be done.

PART 1:

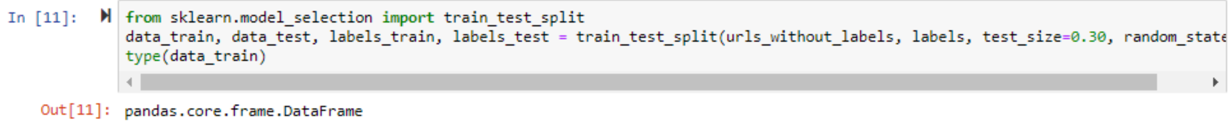
Train data and test data has been created using random forest classifier in anaconda. Then Jupyter notebook helps in displaying the data.

Following is the dataset of generated Legitimate and Phishing Site:



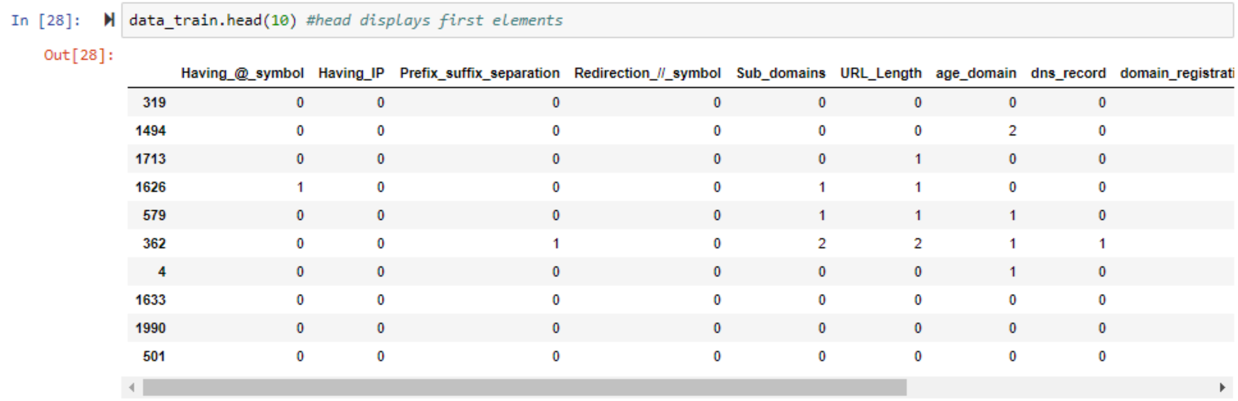
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Domain	Having_@	Having_IP	Path	Prefix_suf	Protocol	Redirectic	Sub_dom	URL_Lengt	age_dom	dns_recor	domain_r	http_toke	label	statistical_	tiny_url	web_traffi
2	www.liqu	0	0	/	0	http	0	0	0	0	0	1	0	0	0	0	2
3	www.onli	0	0	/	0	http	0	0	0	0	0	1	0	0	1	0	1
4	www.cere	0	0	/~nekoi/s	0	http	0	1	0	1	0	1	0	0	0	0	0
5	www.gale	0	0	/kmh/	0	http	0	0	0	0	0	0	0	0	0	0	0
6	www.fanv	0	0	/	0	http	0	0	0	1	1	1	0	0	1	0	1
7	www.anir	0	0	/	0	http	0	0	0	0	0	1	0	0	1	0	1
8	www2.11	0	0	/~mb1996	0	http	0	1	0	1	0	1	0	0	0	0	2
9	archive.rh	0	0	/fritters/y	0	http	0	2	0	0	0	1	0	0	0	0	2
10	www.free	0	0	/	0	http	0	0	0	0	0	1	0	0	0	1	2
11	www.cute	0	0	/	0	http	0	0	0	2	0	0	0	0	0	0	2
12	www.tare	0	0	/	0	http	0	0	0	2	0	2	0	0	0	0	2
13	www.inte	0	0	/users/po	0	http	0	2	2	0	0	1	0	0	0	0	1
14	darkkamir	0	0		0	http	0	0	0	1	1	1	0	0	1	0	1
15	www.iei.r	0	0	/~bkos1/v	0	http	0	2	0	0	0	1	0	0	1	0	1
16	www9.kir	0	0	/fetish/he	0	http	0	0	0	2	0	2	0	0	0	1	0
17	www.jaso	0	0	/	0	http	0	0	0	0	0	1	0	0	0	0	1
18	www.geo	0	0	/kaseycha	0	http	0	2	0	2	0	2	0	0	0	0	2
19	www.ang	0	0	/journal/c	0	http	0	2	2	0	0	0	0	0	0	0	0
20	e.webring	0	0	/hub	0	http	0	0	2	0	0	0	0	0	0	0	2
21	www.nem	0	0		0	http	0	0	0	1	1	1	0	0	1	0	1
22	j-heaven.t	0	0	/library.ht	1	http	0	2	0	0	0	0	0	0	0	0	1
23	www.ang	0	0	/poetry/n	0	http	0	0	0	0	0	0	0	0	0	0	0

Dividing the data into data_train and data_test:



```
In [11]:  from sklearn.model_selection import train_test_split
          data_train, data_test, labels_train, labels_test = train_test_split(urls_without_labels, labels, test_size=0.30, random_state
          type(data_train)

Out[11]:  pandas.core.frame.DataFrame
```

Training Data:

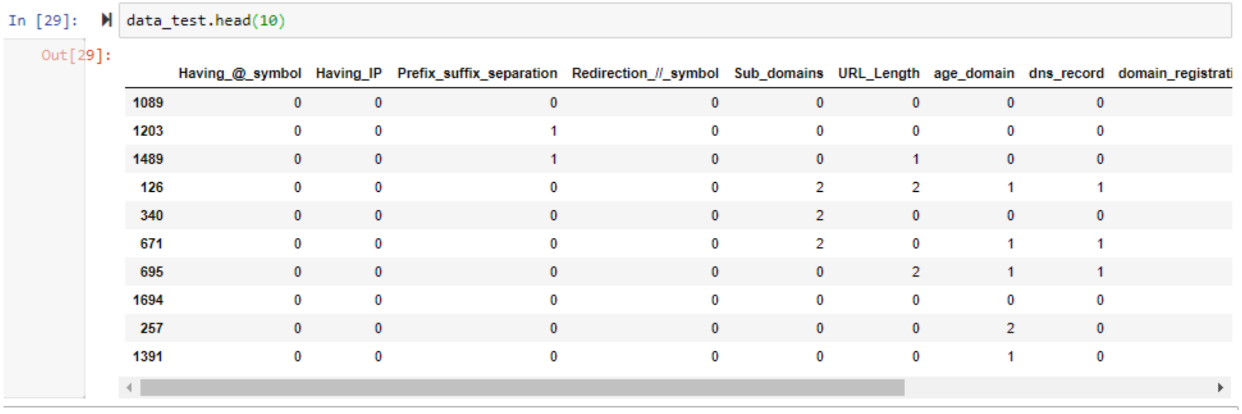


In [28]:
```
data_train.head(10) #head displays first elements
```

Out[28]:

	Having_@_symbol	Having_IP	Prefix_suffix_separation	Redirection_//_symbol	Sub_domains	URL_Length	age_domain	dns_record	domain_registrati
319	0	0	0	0	0	0	0	0	
1494	0	0	0	0	0	0	2	0	
1713	0	0	0	0	0	1	0	0	
1626	1	0	0	0	1	1	0	0	
579	0	0	0	0	1	1	1	0	
362	0	0	1	0	2	2	1	1	
4	0	0	0	0	0	0	1	0	
1633	0	0	0	0	0	0	0	0	
1990	0	0	0	0	0	0	0	0	
501	0	0	0	0	0	0	0	0	

Testing Data:

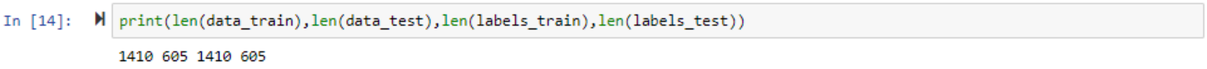


In [29]:
```
data_test.head(10)
```

Out[29]:

	Having_@_symbol	Having_IP	Prefix_suffix_separation	Redirection_//_symbol	Sub_domains	URL_Length	age_domain	dns_record	domain_registrati
1089	0	0	0	0	0	0	0	0	
1203	0	0	1	0	0	0	0	0	
1489	0	0	1	0	0	1	0	0	
126	0	0	0	0	2	2	1	1	
340	0	0	0	0	2	0	0	0	
671	0	0	0	0	2	0	1	1	
695	0	0	0	0	0	2	1	1	
1694	0	0	0	0	0	0	0	0	
257	0	0	0	0	0	0	2	0	
1391	0	0	0	0	0	0	1	0	

Length of Training and Testing Data:

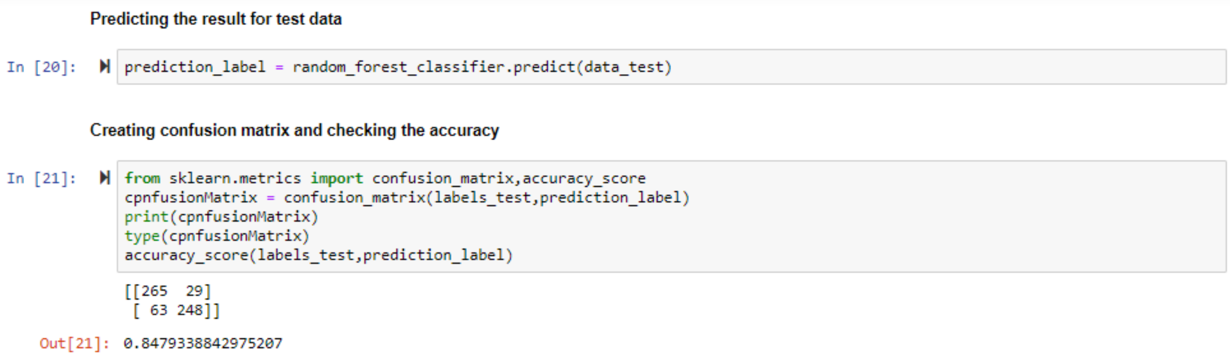


In [14]:
```
print(len(data_train),len(data_test),len(labels_train),len(labels_test))
```

```
1410 605 1410 605
```

PART 2:

Creating confusion matrix and accuracy:



Predicting the result for test data

In [20]:
```
prediction_label = random_forest_classifier.predict(data_test)
```

Creating confusion matrix and checking the accuracy

In [21]:
```
from sklearn.metrics import confusion_matrix,accuracy_score
cpnfusionMatrix = confusion_matrix(labels_test,prediction_label)
print(cpnfusionMatrix)
type(cpnfusionMatrix)
accuracy_score(labels_test,prediction_label)
```

```
[[265  29]
 [ 63 248]]
```

Out[21]:
```
0.8479338842975207
```

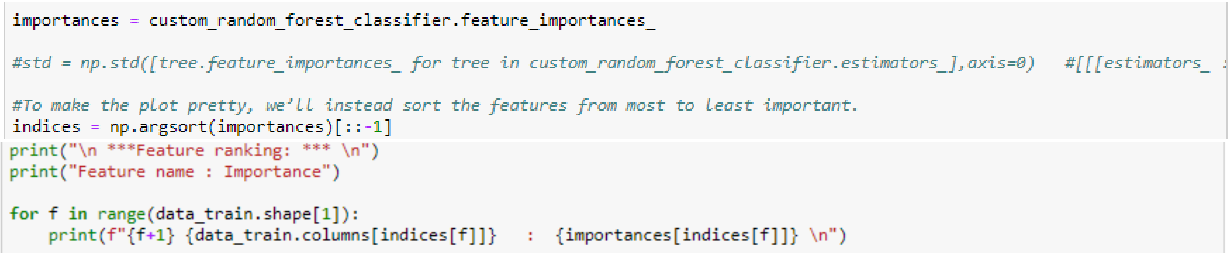
Accuracy can be calculated using following formula:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FN} * 100\%$$

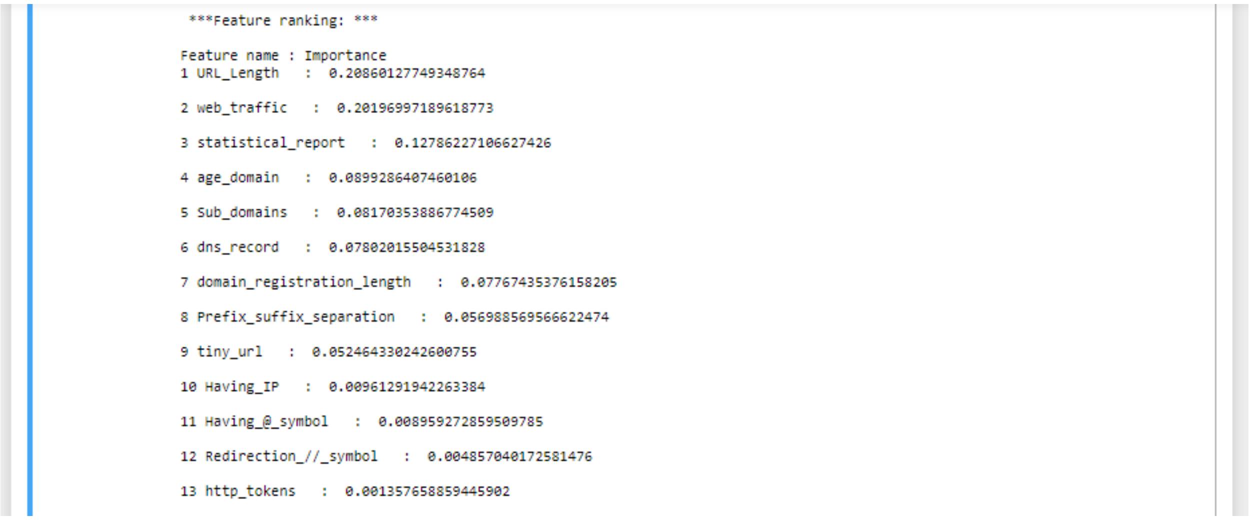
$$Accuracy = \frac{265 + 248}{265 + 29 + 63 + 248} * 100\%$$

$$Accuracy = 84.79338842975207$$

Feature Importance:

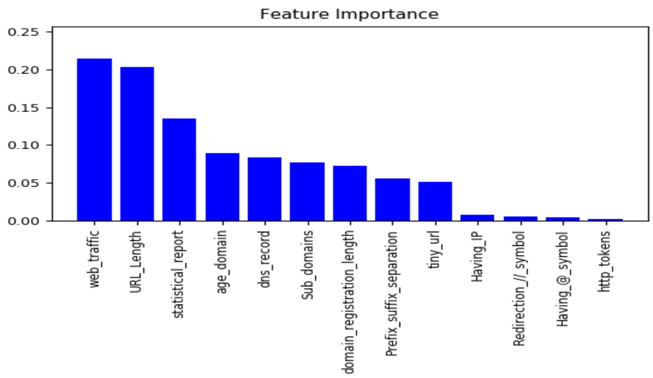


```
importances = custom_random_forest_classifier.feature_importances_

#std = np.std([tree.feature_importances_ for tree in custom_random_forest_classifier.estimators_],axis=0)   #[[[estimators_ :

#To make the plot pretty, we'll instead sort the features from most to least important.
indices = np.argsort(importances)[::-1]
print("\n ***Feature ranking: *** \n")
print("Feature name : Importance")

for f in range(data_train.shape[1]):
    print(f"{f+1} {data_train.columns[indices[f]]}   :  {importances[indices[f]]} \n")
```



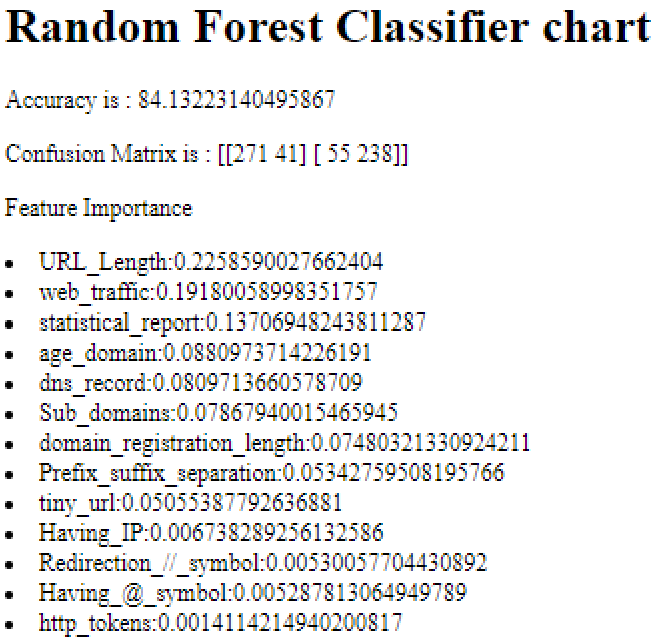
```
 ***Feature ranking: ***

Feature name : Importance
1 URL_Length   :  0.20860127749348764

2 web_traffic   :  0.20196997189618773

3 statistical_report   :  0.12786227106627426

4 age_domain   :  0.0899286407460106

5 Sub_domains   :  0.08170353886774509

6 dns_record   :  0.07802015504531828

7 domain_registration_length   :  0.07767435376158205

8 Prefix_suffix_separation   :  0.056988569566622474

9 tiny_url   :  0.052464330242600755

10 Having_IP   :  0.00961291942263384

11 Having_@_symbol   :  0.008959272859509785

12 Redirection_//_symbol   :  0.004857040172581476

13 http_tokens   :  0.001357658859445902
```

Feature Importance Graph:



Following figure 1,2,3 gives the accuracies of different data sets that are taken from google and iterated. When the collected different data sets and iterated.

Data Set - 1



Random Forest Classifier chart

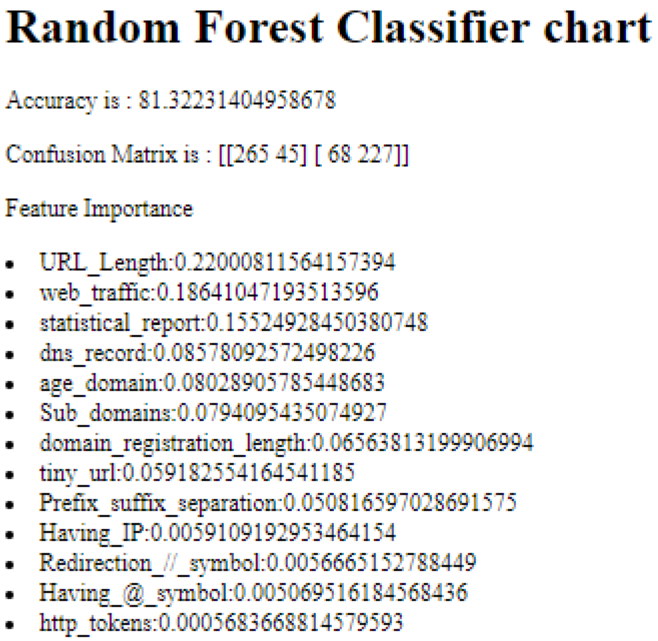
Accuracy is : 84.13223140495867

Confusion Matrix is : [[271 41] [55 238]]

Feature Importance

- URL_Length:0.2258590027662404
- web_traffic:0.19180058998351757
- statistical_report:0.13706948243811287
- age_domain:0.0880973714226191
- dns_record:0.0809713660578709
- Sub_domains:0.07867940015465945
- domain_registration_length:0.07480321330924211
- Prefix_suffix_separation:0.05342759508195766
- tiny_url:0.05055387792636881
- Having_IP:0.006738289256132586
- Redirection_//_symbol:0.00530057704430892
- Having_@_symbol:0.005287813064949789
- http_tokens:0.0014114214940200817

Data Set - 2



Random Forest Classifier chart

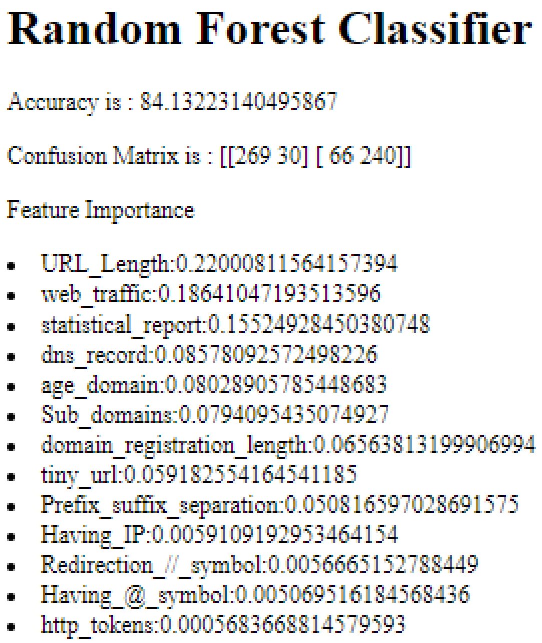
Accuracy is : 81.32231404958678

Confusion Matrix is : [[265 45] [68 227]]

Feature Importance

- URL_Length:0.22000811564157394
- web_traffic:0.18641047193513596
- statistical_report:0.15524928450380748
- dns_record:0.08578092572498226
- age_domain:0.08028905785448683
- Sub_domains:0.0794095435074927
- domain_registration_length:0.06563813199906994
- tiny_url:0.059182554164541185
- Prefix_suffix_separation:0.050816597028691575
- Having_IP:0.0059109192953464154
- Redirection_//_symbol:0.0056665152788449
- Having_@_symbol:0.005069516184568436
- http_tokens:0.0005683668814579593

Data Set – 3



Random Forest Classifier chart

Accuracy is : 84.13223140495867

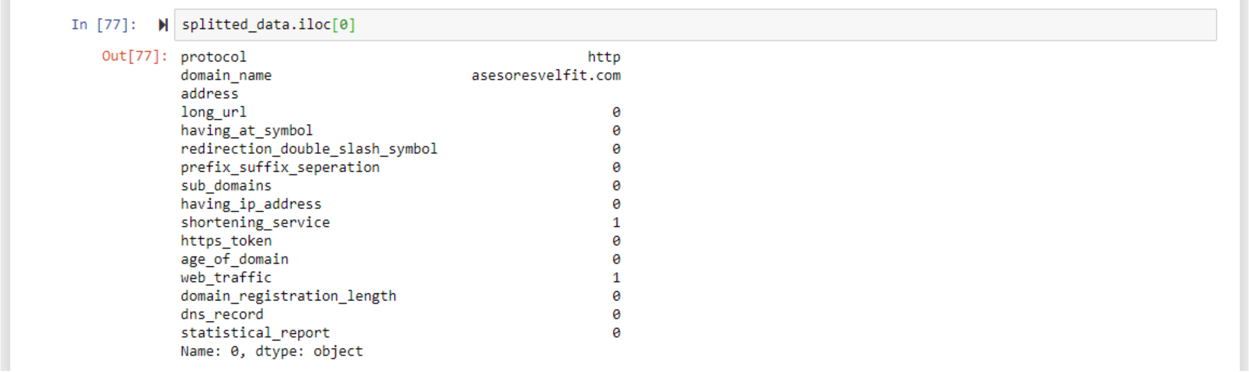
Confusion Matrix is : [[269 30] [66 240]]

Feature Importance

- URL_Length:0.22000811564157394
- web_traffic:0.18641047193513596
- statistical_report:0.15524928450380748
- dns_record:0.08578092572498226
- age_domain:0.08028905785448683
- Sub_domains:0.0794095435074927
- domain_registration_length:0.06563813199906994
- tiny_url:0.059182554164541185
- Prefix_suffix_separation:0.050816597028691575
- Having_IP:0.0059109192953464154
- Redirection_//_symbol:0.0056665152788449
- Having_@_symbol:0.005069516184568436
- http_tokens:0.0005683668814579593

URL Prediction:

Example: Let us consider a phishing link http://asesoresvelfit.com/media/datacredito.co/ and run the features. Following are the features results:



```
In [77]:  splitted_data.iloc[0]

Out[77]:  protocol                                      http
          domain_name                      asesoresvelfit.com
          address
          long_url                                         0
          having_at_symbol                                 0
          redirection_double_slash_symbol                  0
          prefix_suffix_seperation                         0
          sub_domains                                      0
          having_ip_address                                0
          shortening_service                               1
          https_token                                      0
          age_of_domain                                    0
          web_traffic                                      1
          domain_registration_length                       0
          dns_record                                       0
          statistical_report                               0
          Name: 0, dtype: object
```

We can see that above link's result is phishing because of shortening service and web traffic features.

From the above dataset, let us consider the feature importance and calculate the prediction. Feature importance of web traffic is 0.20196997189618773 and shortening service is 0.0524643 30242600755 out of 13 features.

As per the feature importance prediction level for the above site web traffic + shortening servic es i.e. 25%, which is not reliable probability to decide whether the website is phishing or not. Probability level should at least be 40% to predict whether the website the phishing or not.

V. CONCLUSION

In this project, a unique approach is implemented to find out the accuracy of the phishing websites which is already detected by pattern recognition algorithm by using random forests as the classification algorithm with the help of Python. Here, it is empirical demonstrated that which of the features are the most suitable for detection of phishing websites. As per the research, performance metrics proved the accuracy level of the random forest to be the highest around 95% and thus Random Forests were chosen for classification. This approach has used a wide range of metrics, including true positives, true negatives, false negatives, etc. for analysis purposes thus giving a clear view on the performance and accuracy each time the detection takes place. There is no single solution to phishing till now and with the upcoming technology, the type and number of phishing attacks are expected to increase. For these, the browsers have to be made capable enough to setup methods that detect and warn of potential phishing attacks.

VI. FUTURE WORK

Future work will aim to develop a system that can learn by itself about new types of phishing attacks by adding a more enhanced feature to the detection process. The scope of this approach not only helps in adding more enhanced features but also updating the existing features to improve its importance level to make detection more efficient and reduce the false positive rate to a large extent. Another further work should include deploying this approach into a web extension to make the detection more robust to the user.

VII. REFERENCES

[1] Journal of Applied Mathematics Volume 2014, Article ID 425731, 6 pages http://dx.doi.org/10.1155/2014/425731

[2] Must-Know Phishing Statistics 2018. (n.d.). Retrieved April 27, 2019, from https://blog.alertlogic.com/must-know-phishing-statistics-2018/

[3] A. Subasi, E. Molah, F. Almkallawi and T. J. Chaudhery, "Intelligent phishing website detection using random forest classifier," *2017 International Conference on Electrical and Computing Technologies and Applications (ICECTA)*, Ras Al Khaimah, 2017, pp. 1-5.doi: 10.1109/ICECTA.2017.8252051

[4] M. Aburrous, M. A. Hossain, K. Dahal, and F. Thabtah, “Intelligent phishing detection system for e-banking using fuzzy data mining,” Expert Syst. Appl., vol. 37, no. 12, pp. 7913–7921, 2010.

[5] M. He et al., “An efficient phishing webpage detector,” Expert Syst. Appl., vol. 38, no. 10, pp. 12018–12027, Sep. 2011.

[6] M. S. Arade, P. Bhaskar, and R. Kamat, “Antiphishing model with url & image based webpage matching,” Int. J. Comput. Sci. Technol. IJCST, vol. 2, no. 2, pp. 282– 286, 2011.

[7] H. Shahriar and M. Zulkernine, “Trustworthiness testing of phishing websites: A behavior model-based approach,” Spec. Sect. SS Trust. Softw. Behav. SS Econ. Comput. Serv., vol. 28, no. 8, pp. 1258–1271, Oct. 2012.

[8] M. I. A. Ajlouni, W. Hadi, and J. Alwedyan, “Detecting phishing websites using associative classification,” Image (IN), vol. 5, no. 23, 2013.

[9] P. A. Barraclough, M. A. Hossain, M. A. Tahir, G. Sexton, and N. Aslam, “Intelligent phishing detection and protection scheme for online transactions,” Expert Syst. Appl., vol. 40, no. 11, pp. 4697–4706, Sep. 2013.

[10]G. Ramesh, I. Krishnamurthi, and K. S. S. Kumar, “An efficacious method for detecting phishing webpages through target domain identification,” Decis. Support Syst., vol. 61, pp. 12–22, May 2014.

[11]R. M. Mohammad, F. Thabtah, and L. McCluskey, “Intelligent rule-based phishing websites classification,” IET Inf. Secur., vol. 8, no. 3, pp. 153–160, 2014.

[12]N. Abdelhamid, A. Ayesh, and F. Thabtah, “Phishing detection based associative classification data mining,” Expert Syst. Appl., vol. 41, no. 13, pp. 5948–5959, 2014.

[13]D. Zhang, Z. Yan, H. Jiang, and T. Kim, “A domain-feature enhanced classification model for the detection of Chinese phishing e-Business websites,” Inf. Manage., vol. 51, no. 7, pp. 845–853, 2014.

[14]M. Moghimi and A. Y. Varjani, “New rule-based phishing detection method,” Expert Syst. Appl., vol. 53, pp. 231–242, 2016.

[15]Y. Li, R. Xiao, J. Feng, and L. Zhao, “A semi-supervised learning approach for detection of phishing webpages,” Opt.-Int. J. Light Electron Opt., vol. 124, no. 23, pp. 6027–6033, 2013.

[16]G. A. Montazer and S. ArabYarmohammadi, “Detection of phishing attacks in Iranian e-banking using a fuzzy–rough hybrid system,” Appl. Soft Comput., vol. 35, pp. 482–492, 2015.

[17]Y. Li, L. Yang, and J. Ding, “A minimum enclosing ball-based support vector machine approach for detection of phishing websites,” Opt.-Int. J. Light Electron Opt., vol. 127, no. 1, pp. 345–351, 2016.

[18]S. Parekh, D. Parikh, S. Kotak and P. S. Sankhe, "A New Method for Detection of Phishing Websites: URL Detection," *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, Coimbatore, 2018, pp. 949-952.doi: 10.1109/ICICCT.2018.8473085