Text Analysis and Machine Learning Approach to Phished Email Detection

Olasehinde Olayemi Oladimeji
Department of Computer Science,
Federal Polytechnic, Ile Oluji, Ondo State, Nigeria

ABSTRACT
Phishing;, an identity theft of sensitive information poses a serious challenge to security of personal information, it has worrisome effect on countless number of internet users bringing about a huge financial demand on business and victims alike. Text mining is a branch of Data mining used in analyzing large volume of unstructured text data in order to extract meaningful information from it, Machine learning (ML) is an aspect of artificial Intelligence (AI) that uses the method of data mining to find out new or existing characteristics from a set of gathered data which can be relevant for classification. Machine learning methods has been found to achieve much better result than other phished email detection techniques such as blacklists, visual similarity and heuristic techniques. In this work, text mining of phished and ham emails were carried out, three machine learning techniques:- Naive Bayes, K-Nearest Neighbor and Support Vector Machine were used in identifying phished email on a standard analyzed phished email and Ham corpora. From the result, Naive bayes was found to have highest classification accuracy of 99.0% as against the other two machine learning techniques SVM (98.6%) and KNN (96.9%).

Keywords
Identity theft, Text mining, Machine Learning, Sensitive information

1. INTRODUCTION
Text mining is a branch of Data mining concerned with exploring and analyzing large amounts of unstructured text data in order to extract meaningful information from it, text mining tasks includes text classification, text rearrangement, text clustering and text summarization to more some few, text Classification is a supervised machine learning techniques that learn from text documents dataset containing label to build a model for classification and prediction activities. Phished email classification is an example of document classification task which involves classifying an email as either phished or non-phished (Ham) email using machine learning algorithm. Phished email poses a serious challenge to security of personal information, which has caused huge financial lost to the victims of the attack.

Phishing is an identity theft attacks that tricks victims to disclose sensitive information such as passwords, BVN number, bank account number, ATM card details, via a fake website or a spoofed email [1], the Spoofed emails used for phishing presumably comes from a trust worthy individual and it direct the victim to a fake website that looks very genuine [2]. Figure 1 shows an examples of a spoofed email purportedly sent from GT Bank to one of the author, to tricked him to visit a faked GT bank fraudulent websites through links provided in the email, this fraudulent websites mimic the look of the genuine and authentic GT bank website. The email read " We wish to inform you that your token will soon expire and you may not validate transaction with the token again. Kindly follow the reference link below to synchronize your token device and your account will be link to the token. https://ibank.gtbank.com/689fd%2c15cefeccf458024//login.aspx?tokendevice"

Figure 2 shows an example of such the fraudulent website which the link in figure 1 directs victim into in order to collect the victim's sensitive banking information

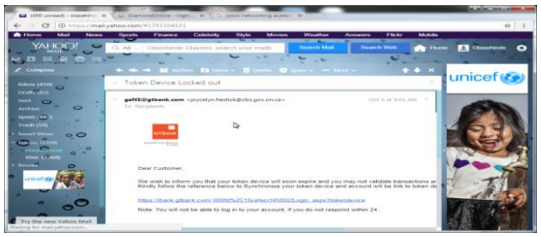

Figure 1: A spoofed email purportedly sent from GT Bank

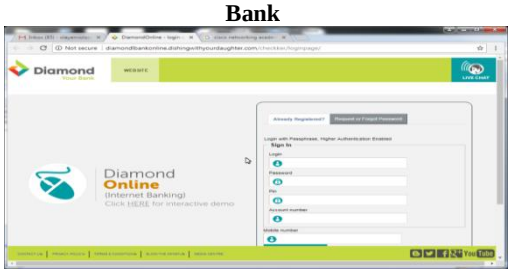

Figure 2: Snapshot of a Fraudulent website used for collecting victim's sensitive information

Checkmating phishing emails is one of the challengers confronting the internet users owing to is high impact on continuous online transaction.. In March, 2006, the Antiphishing working group reported that there were about 18,480 unique phishing attacks as well as 9666 phishing sites respectively, the phishing attacks have worrisome effect on countless number of internet users bringing about a huge financial demand on business and victims alike[3]. In April 2004, a research carried out by Gartner submitted that information supplied to spoofed websites resulted in direct losses for Banks in the United States and credit card issuers to the tune of $1.2 billion [19]. Phishing has thus become a major threat to users of the internet and businesses alike. According to the Anti-Phishing Working Group (APWG) reports of 2014 and 2015 [4], [5], the number of unique phishing e-mail reports received from users has increased tremendously from 68270 e-mails in October 2014 to 106421 e-mails in September 2015 that makes phishing detection one of the hot research topics. APWG, Phishing Activity Trends Report [3], advise computer users as way of prevention against phishing attacks to;

a) be suspicious of any email with urgent requests for personal financial information

b) avoid filling out forms in email messages that ask for personal financial information

The stages of a general phishing attack is presented summarized in figure 3

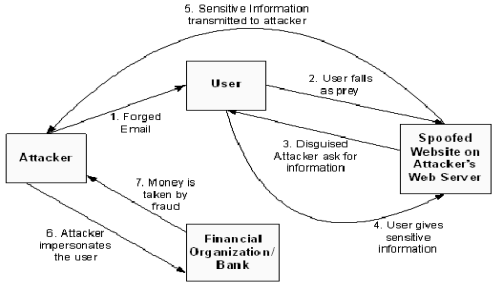

Figure 3: Stages of Phishing Attacks (Sources: [18] Biju et al (2006); Analysis of phishing attacks and countermeasure)

Machine learning (ML) is an aspect of artificial Intelligence (AI) that uses the method of data mining to find out new or existing characteristics from a set of gathered data which can be relevant for classification. In this work, three machine learning techniques:- Naive Bayes, K-Nearest Neighbor and Support Vector Machine are used in identifying phished email on a standard phished email and Ham corpora.

We presented the reviews of related literatures in section 2.1, data set used for this work in section 3.1, the text preprocessing procedures in section 3.2, word embedding vector formation in section 3.3, a detailed description of the machine learning methods in section 3.4, our result findings and conclusion with recommendations were presented in section 4.0 and 5.0 respectively.

2. LITERATURE REVIEW

Phishing attacks are prominently perpetrated via sending of emails, These messages often use a sense of urgency (such as the threat of account suspension) to motivate the user to take action. Recently, there have been several new social engineering approaches to deceive unsuspecting users [6]. These emails usually contain certain keywords that are capable of being used to classify them as either phished or not phished, It is obvious that very few phishing email fitters have been developed over the years whereas, for spam emails, there are many existing email fitters. Blacklisting [7], visual similarity [8], heuristic [9], and Machine Learning [8], are some of the detection techniques for phishing being used by many of the mail fitters developed, Among the few literatures that had addressed the problem of phishing, Machine learning filter outperform every other email filter types in terms of classification accuracy. [10] proposed a new method for detecting phishing emails by incorporating features specific to phishing. [11] presented a comparison of machine learning techniques for phishing detection. The work in [12], used local DNS poisoning attacks at wireless access points to circumvent security of toolbars and phishing filters in a distributed mobile environment using machine learning approach, the target victims were tricked to accepted the information about the faked phished websites as a legitimate one

PhishCatch is a heuristic algorithm proposed by [13] which performs header, link and a cursory text analysis (scanning for the presence of certain text filters) of incoming emails. [14] reviewed the existing literature on the phishing email detection and developed a detection model, that classified phishing messages into two categories: flash and non-flash attack categories, and further classified phishing features into transitory and pervasive.

3. DATASET DESCRIPTION

The dataset used for this research are the Fraudulent e-mails which are phished emails corpora [15], and the Ham public mail corpus provided by spam assassin project [3], the fraudulent e-mails (known as 419 emails, in the Nigeria content) contains criminally deceptive information, usually with the intent of convincing the recipient to give the sender a large amount of money. This dataset is a collection of more than 2,500 "Nigerian" Fraud Letters, dating from 1998 to 2007. 2,500 corpus were selected from the fraudulent corpus and 3,000 Ham corpus were selected from the assassin project, Table 1 shows the composition of the two email corpus used in this work. All the headers information such as; sender email, subject, CC, BCC were removed, only the body (content) of the email were used for the analysis,

Table 1, The Composition of the Datasets

	Fraudulent (Phished) Email	Ham (non-Phished) Email	Total
Number	2500	3000	5500
Percentage	45.45%	54.54%	100%

3.1 Text Preprocessing

Texts such emails are unstructured form of all the available data, various types of noise are present in it and the data is not readily analyzable without any pre-processing. The entire process of cleaning and standardization of text, making it noise-free and ready for analysis is known as text preprocessing. The text preprocessing was carried out on both the training and test email datasets, it involves four stages;

a) Removal of Noise (Stop Words)

b) Lexicon Normalization (Stemming and Lemmatization)

c) Removal of Non word

d) Word Standardization

3.1.1 Abstraction of Stop Words (Noise)

All piece of text which are not relevant to the context of the email are refers to as stop words, words such as "and", "the", "in", "of", "is" etc are very common in text message and they are not very important in determining if an email is phished or not, hence they were excluded from the content of email, this is done by creating a dictionary of noisy entities, and iterate the text object by words, eliminating those words which are present in the noise dictionary. The email corpus, *"Follow the* link below to reset your account to avoid disconnection " will be turned into "Follow link below reset account avoid disconnection " after stop words have been removed from it.

3.1.2 Lexicon Normalization

Normalization is a leadway step for feature engineering with text because it changes the high dimensional feature to a low dimensional space feature, which is suitable for machine learning model building. Multiple representations of a single

word is another type of textual noise that has to be normalized, For instance, the word 'play' has variants such as: Play, Player, Played, Plays and playing. These words are contextually similar but have different meanings. Thus, the most widely used lexicon for normalization practices, remains stemming and lemmatization.

Stemming is the process of breaking down words to their base forms with the objective of reducing related words to their roots as though they have not been extracted from a dictionary. It deals with the removal of certain prefix (such as "ing", "ly", "es", "s", "ed", "tion") from a word without a preservation or loss of the semantic implication of such words.Lemmatization is the procedure for breaking down of a group of words into the lemma or dictionary form. It takes into consideration things like word class, semantics, contextual meaning of such words etc. before breaking them down to their roots. It is an ordered procedure of breaking down or deconstructing a word to its root or base forms. It makes use of vocabularies which emphasize the dictionary importance of words and morphological analysis which emphasizes word formation alongside structure and grammar. Thus, the inflected forms of a word are subject to analysis as a single word. for example "include", "includes", "including" and "included" would be represented as "include", lemmatization unlike stemming, preserves the context and the meaning of the sentence being normalized, Python's Natural language Toolkit (NLTK) library was used to implement the lexical normalization of our dataset

3.1.3 Removal of Non words

Non words like punctuation marks, hash tag, special characters, single character, non alphabetic characters, etc. were also removed from the email text, for convenient and simplicity sake, this operation was carried out after the creation of the dictionary, the python code in figure 4 was used for removing all non words characters from the email documents

```
item_to_remove = dictionary.keys()
        for item in item_to_remove:
if item.isalpha ( )== false
del  dictionary[item]
elif len(item) == 1
del  dictionary[item]
```

Fig 4: Python code for non words, character removal

3.1.4 Word Standardization

Email corpus do contained Words or expressions which are not established in any lexical dictionary of worthwhile standard are contained in the email corpus and as such, such items are not registered in the search engines (of electronic dictionaries) and the models. Examples include: antonyms, abbreviation, informal utterance and slangs. This type of disruption can be curtailed with the use of regular expressions and manually prepared data dictionaries as words can be looked up in a dictionary as a replacement for register of the social media vocals

3.2 Words Embedding

Embedding converts words into numbers many Machine Learning algorithms are not capable of processing text document in their raw form, they are capable of processing numbers as inputs. Word embeddings refer to organized texts which are converted into numbers. With the high volume of data featured in a text format, it becomes pertinent to extract relevant knowledge and structure them up for use. And with the, it is imperative to extract knowledge out of the huge amount of data that is present in the text format and build applications from it. Word embedding map a word using a dictionary to a vector. for the purpose of our work, frequency based count vector embedding was implemented on the email corpus. Consider a Corpus C of D documents {d1,d2…..dD} and N unique tokens extracted out of the corpus C. The N tokens will form our dictionary and the size of the Count Vector matrix M will be given by D X N. Each row in the matrix M contains the frequency of tokens in document D(i). for example,

let corpus D1 is: "account will be closed",

Corpus D2 is " account pin has expired."

The dictionary will create a list of an outline of unique tokens(words) from the two corpus ;

dictionary=['account','closed','pin','expired','will', 'has']

Table 2: The Vector count matrix representation of Corpus D1 and D2

	Account	Close	Pin	Expire	Will	Has
D1	1	1	0	0	1	0
D2	1	0	1	1	0	1

python function; make_dictionary (preprocessed training dataset) was used to create dictionary of 3000 most frequent unique tokens(words) from the 5500 training corpus dataset , single character, special character and non alphabetic words were removed from the dictionary, this reduce the number tokens in the dictionary to 2425 tokens. our vector count matrix has 5500 rows which denotes the 5500 dataset files and 2426 columns denote 2425 most frequent words in the dictionary and the label class with. from Table 3, the value of the index "kl" will be the number of occurrences of lth word of dictionary in kth file

Table 3: format of the vector count matrix of pre processed training set corpus

K/I	Token1	Token 2	....	Token 2424	Token 2425	Class Label
Corpus 1						1Phished
Corpus2						0 - Ham
						0 - Ham
Corpus 5500						1Phished

3.3 Training the Classifier

The email vector created using word embedding discussed in section 3.3 was used for the training and testing of the classifiers using 10-fold cross validation. Basically, 10-fold cross validation is based on data splitting, part of the data is used for fitting each competing model and the rest of the data is used to measure the predictive performances of the classifiers, the dataset is divided into 10 different parts each part consists of 550 email corpus in the ratio of 250 (the fraudulent phished email corpus): 300 (for and Ham unphished email corpus). 9 of the 10 parts was used to train

the classifier while the 10th part is used to validate (test) the classifier, 10 folds cross validation ensures that the training data is different from the test data, and it presents a very good appraisal or assessment of the generalization error of the classifier., the ten folds cross validation procedure was applied on the embedded vector of the cleaned email corpus dataset using three (3) classifier; Naive Bayes (NB), Support Vector Machine (SVM) and K nearest neigbour (KNN), this process is repeated k times, with a different subset reserved for evaluation (and excluded from training) each time until all the 10 folds has been used for training and for testing

3.3.1 Naive Bayes Classifier

Bayesian classifier works on the dependent events and the probability of an event occurring in the future that can be detected from the previous occurring of the same event [16]. This technique can be used to classify phished e-mails; words probabilities play the main rule here. If some words occur often in phished but not in ham, then this incoming e-mail is probably phished. Naive Bayes is a collection of classification algorithms based on Bayes Theorem given by;

$$p(c \mid x) = \frac{p(x \mid c)\, p(c)}{p(x)} \quad (1)$$

$$p(c \mid X) = p(x_1 \mid c) \times p(x_2 \mid c) \times \cdots \times p(x_n \mid c) \times p(c) \quad (2)$$

Where:

P(c|x) is the posterior probability of class (c, phished or ham) given predictor (x, word vectors).

P(c) is the prior probability of class.

P(x|c) is the likelihood which is the probability of predictor given class.

P(x) is the prior probability of predictor.

Naïve Bayes

1. Calculate probabilities for each attribute, conditional on the class value.
2. Use the product rule to obtain a joint conditional probability for the attributes.
3. Use Bayes rule to derive conditional probabilities for the class variable.Once this has been done for all class values, predict class with the highest probability.

3.3.2 K- Nearest Neighbor Classifier

KNN calculates the distance between given instance to be classified and every other instances in the dataset, the label of the instance with lowest calculated distance is predicted as the given instance class, for K = 3, the lowest three (3) instance will be considered, final selection will be based on simple majority vote, the formula for calculating distance between two given instances is given by;

$$d_E(x, y) = \sum_{i=1}^{N} \sqrt{x_i^2 - y_i^2} \quad (3)$$

Where x: are all the instances of the dataset apart from the instance being classified

y: is instance being classified

N: is the total number of instances in the dataset

3.3.3 Support Vector Machine Classifier

The majority of methods developed to deal with the phishing problem are based on support vector machine (SVM). SVM is known machine learning technique that has been used effectively to solve classification problems [17]. Its popularity comes from the accurate results it produced particularly from unstructured problems like text categorization. Given labeled training data (*supervised learning*), the algorithm outputs an optimal hyperplane which categorizes new examples. In two dimensional space this hyperplane is a line dividing a plane in two parts where in each class lay in either side. SVMs are a generally the most appropriate machine learning techniques for text treatment

.SVM searches for a separating hyperplane, which separates positive and negative examples from each other with maximal margin, in other words, the distance of the decision surface and the closest example is maximal

The equation of a hyperplane is:

$w^r x + b = 0$ (4)

The classification of an unseen test example x is based on the sign of $w^T x + b$. The separator property can be formalized as:

$w^T x_i + b \geq 1$ iff 1 $y_i = +1$(6)

$w^T x_i + b \leq 1$ iff 1 $y_i = -1$ (5)

4. RESULT AND DISCUSSION

Machine learning involves two major phases: the training phase and the testing phase, the predictive accuracy of the classier solely depends on the information gained during the training process; if the information gained (IG) is low, the predictive accuracy is going to be low, but if the IG is high, then the classifier's accuracy will also be high. As stated above, we used 10-fold cross validation, on the three machine learning algorithm; Naive Bayes, KNN and SVM to build a phished email classifiers, Table 4, 5, and 6 shows the results of each of the classifiers confusion matrix performance on the dataset.

Table 4: Classification Confusion Matrix of Naive Bayes Classifier

Naive Bayes	Classified as Phished	Classified as Non Phished (Ham)
Phished Email 2500	TP =2475	FN = 25
Non -Phished (HAM) Email 3000	FP = 30	TN = 2970

Table 5: Classification Confusion Matrix of K Nearest Neigbour Classifier

KNN	Classified as Phished	Classified as Non Phished (Ham)
Phished Email 2500	TP =2424	FN = 76
Non -Phished (HAM) Email 3000	FP = 92	TN = 2908

Table 6: Classification Confusion Matrix of Support Vector Machine Classifier

SVM	Classified as Phished	Classified as Non Phished (Ham)
Phished Email 2500	TP =2466	FN = 34
Non -Phished (HAM) Email 3000	FP = 42	TN = 2953

where

True positive (TP): correct positive classification

False positive (FP): incorrect positive classification

True negative (TN): correct negative classification

False negative (FN): incorrect negative classification

The following most commonly used evaluation metrics in literature were used to evaluate the performance of the three classifiers:

Accuracy

Accuracy (ACC) is the ratio of all correct classification to the total number of instances in the test dataset, it is given by equation 7. An accuracy of 1 implies error rate of 0 and an accuracy of 0 indicate error rate of 1

$$ACC = \frac{TP + TN}{FN + FP + TN + TP} \quad (7)$$

True Positive Rate (Sensitivity/Recall)

True Positive Rate (TPR) is the ratio of correctly predicted positive observations to the all observations in actual class, it is given by equation 8, in other word, it is the proportion of the actual positives that are classified as positive by the model, it is also known as sensitivity

$$Sensitivity = \frac{TP}{TP + FN} \quad (8)$$

True Negative Rate (Specificity)

True Negative Rate (TNR) is the proportion of the actual negatives that are detected as negative by the model, it is also known as specificity and it is given by equation 9

$$Specificity = \frac{TN}{TN + FP} \quad (9)$$

False Positive Rate (FPR) or False Alarm Rate (FAR)

False Positive Rate (FPR) or False Alarm Rate (FAR) is the proportion of the wrongly classifier negative as positive by the model, FPR should be as low as possible to avoid unwanted false alarms. it is given by equation 10

$$FPR = FAR = \frac{FP}{TN + FP} \quad (10)$$

Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations. High precision relates to the low false positive rate. it is given by equation 11

$$\Pr ecision = \frac{TP}{TP + FP} \quad (11)$$

F1 score

F1 Score is the weighted average of Precision and Recall. it takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if the dataset contains an uneven class distribution. Accuracy works best if false positives and false negatives have similar cost. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall. F1 score is given by equation 12

$$FI - Score = \frac{2*(\mathrm{Re}call * \Pr ecision)}{(\mathrm{Re}call + \Pr ecision)} \quad (12)$$

Table 8 shows the summary of the performance of each of the classifiers based on the metric evaluation used in this work.

Table 8: Summary of the classifier performance based on the Evaluation Metric

Classifiers	TP	FP	TN	FN	P	F1 score	ACC
Naïve Bayes	0.99	0.010	0.999	0.0100	0.9880	0.9890	0.9900
KNN	0.97	0.030	0.9999	0.0304	0.9634	0.9665	0.9694
SVM	0.98	0.014	0.9999	0.0136	0.9832	0.9848	0.9861

As shown in table 8, the Naive bayes algorithms has the highest performance with accuracy of 99%, True positive of 99%, False Positive FP rate of 1% and precision of 98.8%, followed by Support vector machine (SVM) algorithm with accuracy of 98.6%, False Positive FP rate of 1.4% and precision of 98.3%,, while KNN has the least performance of 96.9% accuracy, False Positive FP rate of 3.0% and precision of 96.3%,, figure 5 shows the performance comparism of the three algorithm based on misclassification rate, (FP + FN), accuracy and precision

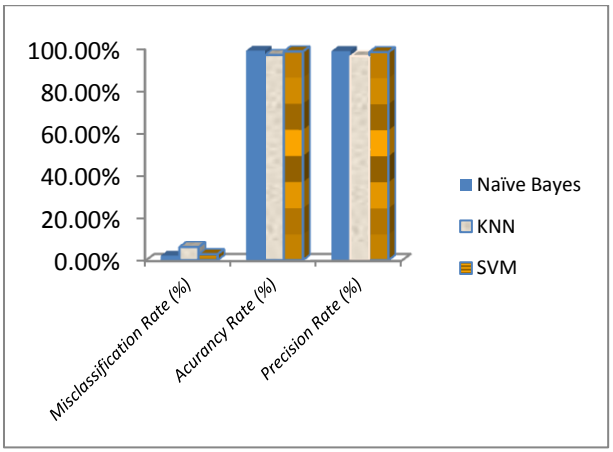

Figure 5. Chart of the Performance comparism of Naive Bayes, KNN and SVM

The computer used in running this test is a 32-bit desktop, core i5, of 2.20 GHz and a RAM and HDD sizes of 6.00 GB. and 500GB respectively, the system is implemented using Python programming language.

5. CONCLUSION AND RECOMMENDATION

Detecting the phishing emails is one of the very crucial problems confronting the internet community because of its high impact on the daily online transactions. In this paper, we have analyzed the various aspects of email phishing attacks both in theory as well as in practice and states some recommendation on how to avoid being a victim of phishing attack. we apply natural language processing to extract useful keywords/token that can be used to determine if an email is a phished mail or not, the vector embedded technique used makes it possible to apply machine learning algorithm to build classification model for detection and classification of phished mail, Naive bayes was found to have highest classification accuracy of 99.0% as against the other two machine learning techniques SVM (98.6%) and KNN (96.9%) used in this work. In the future we intend to create an ensemble of these three methods in other to increase classification performance accuracy and reduce misclassification errors, This system is recommended to be used on mail server to detect and filter phished email and alert the for any incoming phished email.

6. REFERENCES

[1] Beardsley, T., (2005) Phishing Detection and Prevention: Practical Counter-Fraud Solutions, White Paper, 3Com Corporation, Retrieved 15 April 2017, from: http://www.planbsecurity.net/wp/ 503167001_PhishingDetectionandPr evention.pdf

[2] Emigh, A., (2005) "Online Identity theft: Phishing technology, Choke Points and Countermeasures", White paper from Radix Labs. Retrieved 15 April 2017, from: http://www.antiphishing.org/ Phishing-dhs-report.pdf

[3] Anti-Phishing Working Group, (2006), Phishing Activity Trends Report, Retrieved 11th, April 2018 from http://www.antiphishing.org/reports/apwg_report_mar_06.pdf

[4] Anti-Phishing Working Group,(2014) attack trends report, 2014, Retrieved 7th April 2018 from https://docs.apwg.org/reports/ apwg_trends_report_q4_2014.p

[5] Anti-Phishing Working Group, (2015) attack trends report, Retrieved 7th, April 2018 from https://docs.apwg.org/reports /apwg_trends_report_q1-q3_2015.pdf

[6] Anti-Phishing Working Group (2011) Phishing Activity Trends Report, Retrieved 7th, April 2018 from http://www.anti-phishing.org

[7] Prakash P., Kumar M., Kompella R. R., and Gupta M., (2010) PhishNet: predictive blacklisting to detect phishing attacks, in *Proceedings of the IEEE Conference on Computer Communications (IEEE INFOCOM '10)*, IEEE, San Diego, Calif, USA, Ma pp. 1–5.

[8] Bergholz A., Beer de, J., Glahn S., Moens M-F., Paaß G., and Strobel S., (2010) New filtering approaches for phishing email, Journal of Computer Security - EU-Funded ICT Research on Trust and Security, 18(1), P 7-35

[9] Ma L., Ofoghi B., Watters P., and Brown S., (2009) "Detecting phishing emails using hybrid features," in *Proccedings of the Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing (UIC- ATC '09)*, IEEE, Brisbane, Australia, pp. 493–497,

[10] Fette I., Sadeh N., and Tomasic A.,(2007) Learning to detect phishing emails, in Proceedings of the 16th International World Wide Web Conference (WWW '07), Alberta, Canada pp. 649–656,

[11] Abu-Nimeh, S., Nappa, D., Wang, X., and Nair, S.(2007): A comparison of machine learning techniques for phishing detection. In: ACM Proceeding. Anti-phishing Working Group's 2nd Annual eCrime Researchers Summit, pp. 60–69.

[12] Abu-Nimeh (2008) A distributed architecture for phishing detection using Bayesian Additive Regression Trees. Retrieved 16th, April 2018 from http://ieeexplore.ieee.org/ document /4696965/

[13] Yu, W., Nargundkar, S., Tiruthani, N. (2009): Phishcatch-a phishing detection tool. In: 33rd IEEE Int'l Computer Software and Applications Conf., pp. 451–456

[14] Irani, D., Webb, S., Giffin, J., Pu, C. (2008): Evolutionary study of phishing. In: 3rd Anti-Phishing Working Group eCrime Researchers Summit

[15] Radev, D. (2008), CLAIR collection of fraud email, ACL Data and Code Repository, ADCR2008T001, http://aclweb.org/aclwiki or https://www.kaggle.com/rtatman/fraudulent-email-corpus/version/1

[16] Almeida T. A. , Almeida J. and Yamakami A. (2011) Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers, Journal of Internet Services and Applications, Springer, Vol 1, No 11

[17] Yue Zhang, Serge Egelman, Lorrie Cranor, and Jason Hong,(2007) Phinding Phish: Evaluating Anti-Phishing Tools In Proceedings of the 14th Annual Network & Distributed System Security Symposium.

[18] Biju I., Raymond C and Seibu M. J. (2006) Analysis of Phishing Attacks and Countermeasures. Information Security Research Lab, Swinburne University of Technology, Kuching, Malaysia.. Retrieved 08 April 2018, from: https://arxiv.org/ftp/ arxiv/papers/1410/1410.4672.pdf

[19] Litan, A. (2014) Phishing Attack Victims Likely Targets for Identity Theft. Gartner Research (2004). Published: 04 May 2004.