Online Email Phishing Detection using Machine Learning Classifiers

Shubham Mhaske[1], Abhishek Gharge [2], Sarvesh Labre [3], and Jinesh Melvin Y I [4]
[1,2,3,4] Department of Information Technology, PCE, Navi Mumbai, India – 410206

Abstract. Breakthroughs in technology are happening as we speak, but the threat of their misuse is also increasing. Even a tiny amount of exposure within an organization can potentially force the organization out of business. In a digital world, information is the greatest asset. A phishing attack is an attack on the critical information of an individual or an organization. In a phishing attack, the perpetrator uses emails to lure people from different organizations or individuals for using infected URLs, attachments, and offers. The emails contain URLs, sender email information, and reply email information, masked with a legit source to hide the malicious content. Because an individual or the organization receives a vast number of emails every day, it is difficult to detect the infected emails. In such cases, Machine Learning algorithms categorize emails into spam and legitimate mail. A Naive Bayesian network is a supervised Machine Learning algorithm, while it is also an effective way to classify a large number of emails. The Naive Bayesian Classifier is fast in the classification of a large dataset. To further improve the performance, Count Vectorization is applied, and for determining the legitimacy of the sender's email, used Blacklisting algorithm. In this paper, we have analyzed machine learning algorithms for the classification of emails.

Keywords: Naïve Bayesian, Backlisting Algorithm, Spam, Ham,Data Pre-processing, Feature Selection, Phishing, Cyber-attack,

1 Introduction

The surge in the growth of Internet users made email an efficient and powerful communication tool. Although email is a popular tool, it is not immune to cyber-attacks like phishing. Phishing is a type of cyber-attack done using social engineering and deceiving the user to use fake websites to gain sensitive information such as login credentials, bank details, credit-debit card numbers, etc. For the protection against these attacks, conventional web-browsers have a native feature called a blacklist warning system. This feature depends on a database of dictionaries of words or URLs but, the newly added phishing websites escape from getting detected during the process. Today automated methods for filtering emails are necessary. The Machine Learning algorithms are an impressive and intelligent solution given their flexibility in recognizing new phishing websites. So, the system based on machine learning algorithms is better at classifying emails into spam and legitimate. There are numerous algorithms and techniques for email classification. The paper described email classification by the Naive Bayesian Technique (NB). The Naive Bayes classifier belongs to the family of "Probabilistic Classifiers" based on the application of Bayes' theorem with strong independence assumptions between the features [5]. It also describes email features like sender information classification by the Blacklist Classifier Technique. A Blacklist classifier is a technique for discriminating between keywords in a dictionary or URLs. Words or keywords that are blacklisted, trained on comparable data sets. Currently, advancement in technology has increased the intensity of spam emails received by corporate or individuals. Thus, the need for more sophisticated email classification systems is a research issue.

2 Literature Survey

A. Phishing -Malicious Email Detection using Naïve Bayesian Classifier in Data Mining: Phishing attacks can be engineered and are used to target a specific individual or group of people. Targeted Malicious Email (TME) is a phishing attack on the computer network where the general idea is to extract sensitive information from targeted networks. In this context, the conventional anti-malware and antivirus are vulnerable as they focus only on the binary code of the email but fail to scan relevant contextual metadata. Although there are multiple spam filtering methods developed using machine learning algorithms still attackers can violate the routing protocol that can cause a network to become inoperable. As such, Persistent threats and Recipient oriented features support the Naïve Bayesian classifier to demonstrate new techniques that are excellent in detecting Targeted Malicious Emails than conventional email filtering techniques. A dataset of different datasets allowed the

evaluation of this technique. These datasets consist of three classes of emails as follows:
•Targeted Malicious Email (TME).
•Non-Targeted Malicious Email (NTME).
•Evaluation set containing both NTME and TME.
The NTME and TME are used to construct the TME-filter and create new features included for TME detection. The complete process consists of few phases they are
1.Preprocessing of data.
2.Feature extraction using Naïve Bayesian Classifier.
3.Classification as Targeted or Non-targeted Malicious Email [1].

B. Detection of Phishing Websites Using Naïve Bayes Algorithms: The phishing websites masked as legitimate are used to deceive users into delivering their sensitive information like login credentials, bank details, and credit card information. Currently, most URL anti-phishing systems use machine learning algorithms like Naive Bayes, Decision Trees, and Random Forest Algorithm. These algorithms have impressive accuracy rates on their own then this paper discusses using new methodologies along with these algorithms to boost their performance and increase their accuracy rates [2]. Techniques proposed such as Stacking, Bagging, and Boosting help algorithms achieve an accuracy of 97.08%, even though it only investigates the features extracted from URLs. The broad usage of the techniques along with Naive Bayes resulted in accuracy rates after boosting as 85.5137%, after bagging as 89.8004%, and after stacking as 51.8847 [2].

C. An Efficient Email Spam Detection using Support Vector Machine: Emails are now part of our daily life due to their popularity and rapid growth in technology. As emails achieve popularity, this led to an increase in spam. Spam emails are unsolicited electronic mails that are sent in high volumes to gain personal credentials. The conventional mail security systems follow a mechanism wherein Mail Headers get checked and where rules are specified. Each spam email has a signature where the hash value is unique for every message in the Header. Later, these rules and mail headers are checked for a match to detect spam. These approaches fail in filtering spam mails and leave an unwanted residual that affects the user. Thus, to create efficient spam filtering systems, machine learning algorithms such as Support Vector Machine (SVM), Naive Bayesian is used. Support Vector Machine (SVM) belongs to the supervised machine learning algorithm group. SVM works by detecting a hyperplane to classify the dataset into different classes. Then, there is the K Means Clustering that is an unsupervised algorithm that efficiently performs cluster analysis. The final Algorithm discussed in the paper is the Naive Bayes Classifier Algorithm that is a statistical classifier excellent in the classification of emails. These algorithms are tested and benchmarked based on their accuracy, precision, recall, and F-measure [3].

D. An Intelligent Spam Detection Model Based on Artificial Immune System: Most spam filtration systems nowadays use machine learning algorithms in combination with other techniques. While in this paper, an innovative method reveals that our body's immune system can be an inspiration for the spam detection model. In Artificial Immune Systems (AIS), processes from the natural immune system are abstracted and applied in [the field of] science and engineering. In an immune system of a mammal, a mechanism called Thymus produces T-cells that are capable of binding only with non-self (unfamiliar/unmatched) antigens. These T-cells possess a repository of know self-cells through which T-cells get trained during the maturity phase to avoid binding with self-cells. During an accident or break in the system, T-cells actively bind with the non-self-antigens. This unwanted combination of T-cells and non-self-antigens is later dealt with by antibodies. Based on this mechanism, the Negative Selection Algorithm (NSA) is developed. The procedure in the Negative Selection Algorithm begins by producing a set of self-strings. Self-strings are a known pattern that defines the normal state of the system. Then there is the production of detectors that are responsible for binding with non-self-strings. After detecting the non-self-strings, binding occurs with the spam keywords or blacklisted URLs and IPs by detectors. The NSA Algorithm can specify between self or non-self (spam or legit) datasets to build a knowledge base for intelligent systems [4].

3 Proposed Work

3.1 Preliminaries

We propose a spam detection system based on the Naive Bayes classifier and Blacklist classifier. Though decision trees have advantages in numeric domains but they don't work well if lots of features are equally important. The Naïve Bayes classifiers are optimal for independence assumptions. The accelerated processing ability of the Naive Bayes classifier without getting overwhelmed by a large number of emails and increment in performance after combining with blacklist classifier. These are the traits through which a powerful email filtration system can be developed.

3.1.1 Naïve Bayes Classifier

The Naive Bayes Classifier is a probabilistic classifier wherein Bayes' theorem is applied with strong independence assumptions between the features [5]. It is a probability-based classifier that computes classes for a problem instance for predictions. These instances are represented as features in the classifier, and they are considered independent during the computation of a probability. This family of algorithms works efficiently with supervised learning settings. Equation 1 is the mathematical representation of the Naive Bayes Classifier.

$$\Pr(C_k|f) = \frac{\Pr(f|C_k)\Pr(C_k)}{\Pr(f)} \quad (1)$$

Here, f is a set of feature vectors that can be represented as $f = (f_1, f_2, \dots, f_n)$. In C_k, C stands for the class variable for each k possible outcomes. Now, $\Pr(C_k|f)$ is posterior probability, $\Pr(C_k)$ is prior, $\Pr(f|C_k)$ is a likelihood, and $\Pr(f)$ is evidence so, the posterior probability depends on the likelihood of a set of features belonging to class $\Pr(C_k|f)$ where $\Pr(C_k)$ is the prior probability and $\Pr(f)$ is evidence depending on the known feature variables. In a simple example, an object may be considered a cube if all sides have equal dimensions, the edges opposite are parallel, and plane angles are the right angle. Concerning the Naive Bayes Classifier, each of these features contributes independent probability that the object is a cube regardless of any possible correlation between dimensions.

3.1.2 Multinomial Naïve Bayes Classifier

Multinomial Naïve Bayes Classifier is a supervised learning algorithm that is based on a probabilistic nature. The Multinomial Naive Bayes classifier is efficient in the classification of discrete features such as word counts for text classification. It requires more than two integer feature counts..

$$\hat{P}(c) = \frac{N_c}{N} \quad (1)$$

$$\hat{P}(w|c) = \frac{count(w,c)+1}{count(c)+|V|} \quad (2)$$

As per the above equations 1 and 2. The probability of a class is the number of documents with that class over the total number of documents and the likelihood of a word given a class is the word occurring in that class divided by the count of all the words.

3.1.3 Blacklist Classifier

A blacklist classifier is a predictive tool that is helpful for correctly classifying or predicting if a URL or sender content is likely to be malicious or not. Blacklists are used in every corner of the anti-fraud industry to identify if a device, user, or IP address is dangerous. A blacklist is the list of senders, IP addresses, and email addresses that have been previously marked or

flagged as unsafe. It works by checking email addresses or IP addresses of the new incoming emails against the blacklist, and if found on the list, then the email is isolated. So, the blacklists create lists in the database that are browsed constantly over time. Blacklists can be of both small and large scale. For the small-scale blacklists, users have control to will if they don't want to allow email from specific addresses. The use of small-scale blacklists is only if the user receives spam emails from a particular address. The large-scale blacklists, on the other hand, are provided by third parties. In an extensive list like this, users do not contribute. Blacklist classifiers are also used in classifying URLs in emails.

3.2 System Architecture

The email phishing detection system (EPDS) follows multiple phases to conclude the severity of the email. The system mainly consists of two principal phases, data organization, and data evaluation. Organizing consists of data preprocessing, Tokenization, and feature extraction. And the other part consists, training of machine learning algorithms using a training dataset and email classification using these trained models into spam and ham.

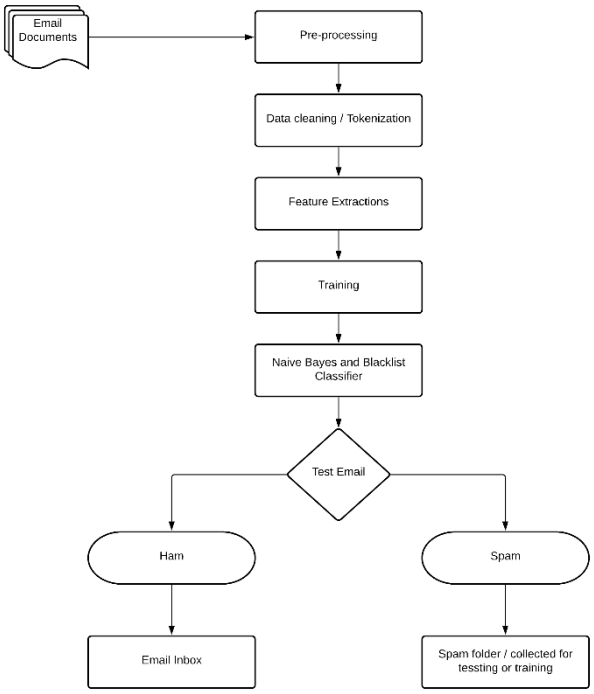

Fig. 1. Proposed system architecture for email classification

3.2.1 Dataset

The dataset used for testing and training the machine learning models is a public dataset collected and presented by the CALO Project [8]. It contains 0.5M messages from about 150 users of Enron organized into folders. The dataset is available at the domain data.world by Brian Ray [9]. The dataset is used for preprocessing, feature extraction, training, testing, and analysis.

3.2.2 Preprocessing

This is the first stage of the EPDS that is executed on the incoming emails. The efficiency of the Email Phishing Detection

System (EPDS) increases with the support of proper preprocessing steps. Preprocessing is the removal of the stop words that are the common words in a language, such as the articles, conjunctions, and prepositions. Web page residuals like HTML tags are also removed during preprocessing. In preprocessing, tokenization is a type of process that segments the content of the emails into individual characters [1]. Since the data arriving at the system is raw, therefore normalization of text is important to increase the uniformity of preprocessing. The process in which text is either converted to upper case or lower case, where punctuations are removed, is called normalization.

3.2.3 Features Extraction

Features is a set of terms that helps a machine learning model to predict at an efficient rate. These terms are a set of words frequenting in a document or email and have importance concerning that document. The extraction of these features is achieved with the help of the vectorization technique. TF-IDF Vectorizers have higher efficiency when combined with probabilistic algorithms. Naïve Bayes achieves a world level tf-idf precision rate of 56% that is the second-highest precision [7].
TF IDF: It is short for term frequency-inverse document frequency, used to evaluate how important a word is in a document. Term frequency is the statistical weighted calculations that how many times a word occurs in a document to the total number of words in the document. And inverse document frequency is about how important the word is. For example, when operating through an email dataset, there is a high probability that words such as "offer" can be present more than 100 times, and the document contains 1000 words then, the term frequency will be 100/1000 = 0.1. And suppose if there are 10000 emails and 200 of them have the word "offer" then, the IDF value is 10000/200 = 50 so, the TF-IDF will be 0.1*50 = 5 [7].

3.2.4 Classification

The features that are converted as vectors in the previous steps are trained using classifiers. These classifiers are constructed using the Naive Bayes machine learning algorithm. The working of the Naïve Bayes and Blacklist classifiers is discussed in the above section (3.1).

3.2.5 Algorithm for Naïve Bayes Classifier

Step 1: Select the email.

Step 2: Preprocess/Extract features using vectorization algorithm.

Step 3: Training the dataset using the Naive Bayesian Classifier.
Calculate posterior probability using Naïve Bayesian of every class.
The highest posterior probability is the result for the prediction.

Step 4: Determining the probability of spam and non-spam emails'
Step 5: Dataset testing.

Step 6: Classification of emails into spam and non-spam.

4 Performance and Results

The Enron dataset used for testing and training confers the total number of spam and ham messages below.

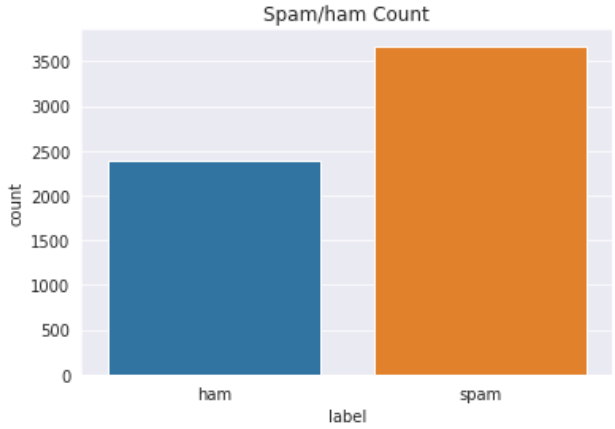

Fig. 2. Spam and ham count in the dataset

Feature extraction allows the classifiers to optimize the predictions, it allows to achieve the efficient classification of data. After preprocessing data, vectorization of data through TF-IDF, these are the following top 33 features with respect to their Term Frequency and Inverse Document frequency.

Table. 1. Features extracted using TF IDF

index	idf	tfidf	term
10186	5.5665	0.124	busi
12852	5.972	0.133	complet
14755	6.6651	0.2969	david
16206	5.9961	0.1336	document
17688	6.3423	0.1413	employe
17745	5.6492	0.1258	end
19888	9.0165	0.2008	forwardrna
20692	4.307	0.1919	gener
21136	4.6857	0.1044	go
23555	5.3152	0.1184	hpl
23577	6.3423	0.1413	hr
25406	5.7023	0.381	issu
25877	6.1261	0.2729	john
26288	6.8764	0.1532	key
27436	3.846	0.0857	like

29689	6.937	0.1545	mid
31016	3.4811	0.0775	need
31346	9.0165	0.2008	nim
32468	8.3233	0.1854	ongo
33939	4.8733	0.1086	perform
34631	5.6321	0.1255	plan
37349	5.2323	0.1165	report
37610	4.5857	0.2043	review
38130	6.8193	0.1519	rnb
38262	7.407	0.165	rnc
39373	9.0165	0.2008	rnoffici
39583	9.0165	0.2008	rnregardsrndelainey
43999	1	0.0223	subject
46132	7.0706	0.1575	transfer
46147	6.5316	0.1455	transit
46963	6.937	0.1545	unit
49227	4.3344	0.0965	work
49721	5.2437	0.2336	year

The dataset used for testing and training contained 777 spam emails, 435 ham emails, and the total number of emails was 1212. When evaluating the model, metrics assist in analyzing the accuracy of the Naïve Bayesian model.

4.1 Metrics used for evaluation of the emails

True Positive (TP): Email is truly spam

True Negative (TN): Email is truly ham

False Positive (FP): Email was ham but classified as spam

False Negative (FN): Email was spam but classified as ham

Accuracy: It states the correctly classified emails out of the complete dataset. It is the ratio of true positives and true negatives to the true positives, true negatives, false positives, and false negatives.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

Precision: Itcalculates the percentage of emails that are categorized as spam and that actually are spam. It is calculated as follows:

$$Precision = \frac{TP}{TP+FP} \quad (2)$$

Recall: It is used to check the retrieval, about what percentage of spam emails classifier labels as spams.

$$Recall = \frac{TP}{TP+FN} \quad (3)$$

F-Score: It is an average of precision and recall. Calculated as:

$$F_{score} = \frac{2*Precision*Recall}{Precision+Recall} \quad (4)$$

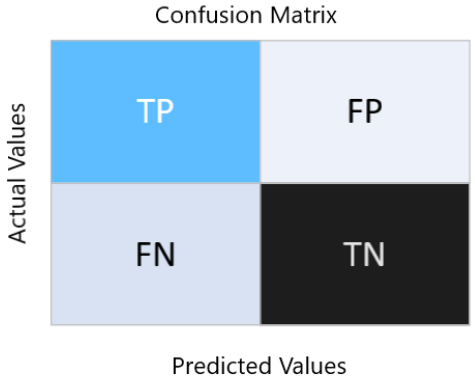

Fig. 2.1 Confusion matrix withreference model

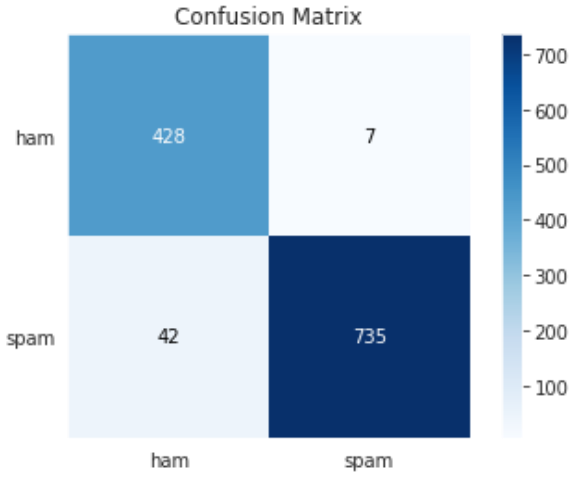

Fig. 2.2 Spam and ham confusion matrix

The Naïve Bayesian model scored an accuracy of 0.959% i.e.,96%. Below are the results for all the metrics associated with the

analysis.

Table. 1.1 Spam and ham results using Naïve Bayes

	precision	recall	f1-score	support
ham	0.91	0.98	0.95	435
spam	0.99	0.95	0.97	777
accuracy			0.96	1212
Macro avg	0.95	0.96	0.96	1212
Weighted avg	0.96	0.96	0.96	1212

5.1 Conclusion

In this paper, we have worked on machine learning classifiers in the context of email phishing detection. We have analyzed different classifiers like the naive Bayes and the blacklist classifiers used for classifying emails. We used TF IDF Vectorizer to extract the features from the dataset, improves the precision of the Naive Bayes classifier. The Naive Bayes Classifier scored an accuracy of 96% with the given dataset. The results produce a precision of 99% for spam that proves how efficiently the classifier can distinguish spam emails. The classification results were impressive in terms of precision, recall, and f-score. The future idea is to combine the Naive Bayes Classifier with other techniques and try a hybrid approach in creating a spam filter for websites, emails, and mobile apps

ACKNOWLEDGMENT

We are honored to express our special thanks of gratitude to our project guide **Prof. Jinesh Melvin Y. I.**, for his expert guidance and support in completing our project.

We would also like to express our sincere thanks to our **Head of the Department**, **Dr. Satishkumar Varma**, for encouragement and cooperation throughout the project.

We are affable indebted to **Principal Dr. Sandeep M. Joshi** for encouraging andallowing us to present this work.

References

[1] Balaji M, Padmanabhan V ,Tharun Kumar J, Mr. Arockia Abins :PhishingMalicious Email Detection using Naïve Bayesian Classifier in Data Mining: VDGOOD Journal of Computer Science Engineering, April 2020

[2] Gyan Kamal, Monotosh Manna: Detection of Phishing Websites Using Naïve Bayes Algorithms: International Journal of Recent Research and Review ISSN 2277–8322, Vol. XI, Issue 4, December 2018

[3] K sai Prasanthi, T Deepika, S Anudeep, M Sai Koushik: An Efficient Email Spam Detection using Support Vector Machine: International Journal

of Innovative Technology and Exploring Engineering (IJITEE) ISSN: 2278-3075, Volume-9 Issue-2, December 2019
[4] Abdul Jabbar Saleh, Asif Karim, Bharanidharan Shanmugam, Sami Azam, Krishnan Kannoorpatti, Mirjam Jonkman and Friso De Boer: An Intelligent Spam Detection Model Based on Artificial Immune System: MPDI, June 2019
[5] Naive Bayes classifier: Wikipedia, last modified March 29, 2021, https://en.wikipedia.org/wiki/Naive_Bayes_classifier
[6] Blacklist Classifier: Bitbucket, last modified October 19, 2017, https://bitbucket.org/tiedemann/blacklist-classifier/wiki/Home
[7] Ravinder Ahuja, Aakarsha Chug, Shruti Kohli, Shaurya Gupta, and Pratyush Ahuja: The Impact of Features Extraction on the Sentiment Analysis: International Conference on Pervasive Computing Advances and Applications – PerCAA 2019, Procedia Computer Science 152 (2019) 341–348
[8] Enron Email Dataset William W. Cohen, MLD, CMU, Last modified May 8 2015, https://www.cs.cmu.edu/~enron/
[9] "Enron Email Dataset" by Brian Ray, tinyurl.com/26b9sy4j
[10] Adwan Yasin and Abdelmunem Abuhasan: AN INTELLIGENT CLASSIFICATION MODEL FOR PHISHING EMAIL DETECTION: International Journal of Network Security & Its Applications (IJNSA) Vol.8, No.4, July 2016
[11] Tiago A. Almeida, Jurandy Almeida, Akebo Yamakami: Spam filtering: how the dimensionality reduction affects the accuracy of Naive Bayes classifiers: © The Brazilian Computer Society 2010 J Internet Serv Appl (2011) 1: 183–200 DOI 10.1007/s13174-010-0014-7, December 2010
[12] Amit Kumar Sharma, Sudesh Kumar Prajapat, Mohammed Aslam: A Comparative Study between Naïve Bayes and Neural Network (MLP) Classifier for Spam Email Detection: International Journal of Computer Applications® (IJCA) (0975 – 8887) National Seminar on Recent Advances in Wireless Networks and Communications, NWNC-2014
[13] Aman Kumar Sharma, Suruchi Sahni: A Comparative Study of Classification Algorithms for Spam Email Data Analysis: International Journal on Computer Science and Engineering (IJCSE) ISSN : 0975-3397 Vol. 3 No. May 2011
[14] V.Christina, S.Karpagavalli, G.Suganya V. Christina et al.: Email Spam Filtering using Supervised Machine Learning Techniques: (IJCSE) International Journal on Computer Science and Engineering ISSN : 0975-3397 Vol. 02, No. 09, 2010, 3126-3129
[15] Ms. Kranti Wanawe, Ms. Supriya Awasare, Mrs. N. V. Puri: An Efficient Approach to Detecting Phishing A Web Using K-Means and Naïve-Bayes Algorithms: International Journal of Research in Advent Technology, E-ISSN: 2321-9637 Vol.2, No.3, March 2014
[16] Harikrishnan NB, Vinayakumar R, Soman KP: A Machine Learning approach towards Phishing Email Detection: CEN-Security@IWSPA 2018
[17] Bin Ning, Wu Junwei, Hu Feng: Spam Message Classification Based on the Naïve Bayes Classification Algorithm: IAENG International Journal of Computer Science, 46:1, IJCS_46_1_05, February 2019
[18] Gurneet Kaur, Er. Neelam Oberai, Gurneet Kaur et al: A REVIEW ARTICLE ON NAIVE BAYES CLASSIFIER WITH VARIOUS SMOOTHING TECHNIQUES: International Journal of Computer Science and Mobile Computing, Vol.3 Issue, October 2014, pg. 864-868