7th National Congress of New Findings in
Electrical Engineering

E-Mail phishing detection using natural language processing and machine learning techniques

Seyyed Rohollah Mirhoseini[1,*] Fatemeh Vahedi[2], Jalal A Nasiri[3]
[1] Computer Department Islamic Azad University North Tehran branch Tehran, Iran.
Email: mirhoseini@alumni.iust.ac.ir
[2] Computer Department Islamic Azad University North Tehran branch Tehran, Iran.
[3] Iranian Research Institute for Information Science and Technology
corresponding author : mirhoseini@alumni.iust.ac.ir

ABSTRACT

Today there are more than 4.39 billion internet users that almost 70 percent of them use social media on mobile devices. Network security is one of the most important aspects to consider while working over the internet. E-mail is one of the most secure medium for online communication and transferring data or messages through the web. This paper illustrates an email spam detection method using natural language processing and machine learning techniques (MLT). Then we present the classification and evaluation results.

Keywords: *Network security, Machine Learning (ML), Natural Language Processing (NLP) and Deep Learning.*

1. INTRODUCTION

Network security is a difficult task and collection of policies, techniques, technologies, and processes that work together to protect the confidentiality, integrity, and availability of computing resources, networks, software programs, and data from attack is a challenging activity [1].

Traditional security methods relies on the static control of security devices deployed on special edges or nodes, such as firewalls, antivirus software, intrusion detection systems (IDSs), and intrusion prevention systems (IPSs) for network security. However, this passive defense methodology is no longer useful for protecting systems against new network security threats because they only need to find vulnerability in the systems needing protection. Furthermore, attackers are becoming more sophisticated, developing advanced persistent threats (APTs), zero-day exploits and malwares that evade security measures which enable them to persist for longer periods without notice [1].

Since the Internet of Things (IoT) such as smart cities, smart buildings, healthcare, smart grids and industrial manufacturing groups are growing rapidly the security and the privacy are becoming more important.

The adversarial attacks can be divided into three categories: attacks in the training stage, testing stage, and model deployment stage [2].

The training stage adversarial attacks refer to the fact that, in the training stage of the target model, the adversaries carry out attacks by modifying the training dataset, manipulating input features or data labels [2].

The testing stage adversarial attacks can be divided into white-box attacks and black-box attacks. In white-box scenarios, the adversaries have access to the parameter, algorithms, and structure of the target model. Whereas in the black-box scenarios, the adversaries cannot obtain information about the target model, but they can train a local substitute model by querying the target model, utilizing the transferability of adversarial samples or using a model inversion method [2].

In natural language processing fields, there are adversarial attacks in machine translation and text generation. In the cyberspace security field, there are adversarial attacks in cloud service, malware detection, and network intrusion detection. Researchers have proposed several adversarial attack defense strategies, which can be divided into three main categories, i.e., modifying data, modifying models, and using auxiliary tools [1].

With the advent of the internet, “phishing” is the most popular way to steal an identity. Same as traditional fishing where the fisherman troll the river in a boat to catch fish, in “phishing”, attackers trolls the Internet using email messages with convincing content as baits to steal users personal information. The email directs the user via a hyperlink to a website owned by criminals that looks very similar to a legitimate website. The user will then be asked to enter

personal and financial information either to update existing information or to purchase a product. In reality, this lets the criminal have access to valuable information which they use to commit fraud or to sell it to a bidder. Phishers can also trick users into downloading malicious codes or malware after they click on a link embedded in the email. This is a useful tool in crimes like economic espionage where sensitive internal communications can be accessed and theft of trade secrets. Phishing has used since 1996, but it has become more common and more sophisticated. Recent phishing attacks occurred in the Gmail system to stole US government officials, contractors, and military personnel emails [3].

Considerable research has been done towards protecting users from phishing attacks. They include firewalls, black listing certain domains and Internet protocol (IP) addresses, spam filtering techniques, client side toolbars, and user education. Each of these existing techniques has some advantages and some disadvantages. For example, existing filters have misclassification rates, the blacklist approach is hard to maintain for every expanding IP address/domain space, while the user ignores client side toolbar warnings [4].

To effectively handle the threat posed by email spams, leading email providers such as Gmail, Yahoo mail and Outlook have employed the combination of different machine learning (ML) techniques such as Neural Networks in its spam filters. These ML techniques can learn and identify spam mails and phishing messages by analyzing loads of such messages throughout a vast collection of computers. Since machine learning can adapt to varying conditions, Gmail and Yahoo mail spam filters do more than just checking junk emails using preexisting rules. They generate new rules themselves based on what they have learnt as they continue in their spam filtering operation [5].

We used email datasets from SpamAssassin corpus. We need to preprocess emails to create a feature matrix with rows being the emails and the columns being the features. Then we used Natural Language Processing methods for preprocessing .Finally by SVM classification method spam emails seperate from non-spam ones.

2. DESCRIPTION OF ATTACKS

2.1 Description of Security

During the data transmission, the communication links should not be susceptible to any type of attack. A potential hacker can target a communication channel, retrieves the secret keys, decrypt them and injects false data in the network. Network security is important just like the security of computers and the encryption of the messages to develop secure network need few points to consider which are listed as below [1]:

- **Confidentiality:** The data in the network persists to be private.
- **Access:** Only authorized users have access to communicate over the network.
- **Integrity:** Ensures the data in transit is not modified and reaches the destination in actual form.
- **Authentication:** Ensures the users in the network are those who they claim to be.
- **Non-repudiation:** Ensures the user doesn't contradict who have used the network.

2.2 Description of Attacks

Attacks are classified into many categories and we need to know them, here we present famous attacks:

Worms: Worms and viruses have a similar property of selfreplicating. But, the former doesn't require a file to replicate itself and propagate throughout the system. Network ware worms and mass-mailing are two main types of worms. A network aware worm chooses a target and typically infects it by a Trojan or other. In mass mailing worms email is a means to infect the target [6].

Viruses: Viruses can infect the files and propagates throughout the system by replicating themselves [6].

Trojans: Trojans seem to be harmless for the system, but carries some malicious intention. They normally transport some payload like a virus [6].

Phishing and Malware attack: Phishing is the most common type used for attacking victims. It is when a malicious party sends a fraudulent email or message masked as a legitimate content. The message goal is to trick the recipient into sharing personal or financial information or clicking on a link that installs malware or leads websites to impersonate real systems to capture sensitive data. For example, a message might come from a bank or other well-known institution with the need to verify your login information [7].

Spam campaign attacks: Spam attack is spam or spam campaigns that embed phishing advertisements in an email or a post on Facebook [7].

Phishing and eavesdropping attacks: Phishing and eavesdropping attacks are carried out to gain personal information and system knowledge. Some attacks like worms, viruses, and Trojans are perpetrated to alter the systems function. DoS is a type of attack in which the system resources are consumed so heavily that it makes the system inoperative [6].

2.3 E-mail attack detection

Content Based Filtering Technique: Content based filtering is usually used to create automatic filtering rules and to classify emails using machine learning approaches, such as Naïve Bayesian classification, Support Vector Machine, K Nearest Neighbour, Neural Networks. This method normally analyses words, the occurrence, and distributions of words and phrases in the content of emails and then uses generated rules to filter the incoming email spams [5].

Case Base Spam Filtering Method: Case base or sample base filtering is one of the popular spam filtering methods. Firstly, all emails both non-spam and spam emails are extracted from each user's email by using collection model. Subsequently, pre-processing steps are carried out to transform the emails using client interface, feature extraction and feature selection, grouping email data, and evaluating the process. The data is then classified into two vector sets. Lastly, the machine learning algorithm is used to train datasets and test them to decide whether the incoming mails are spam or non-spam [5].

Heuristic or Rule Based Spam Filtering Technique: This approach uses existing rules or heuristics to assess a huge number of patterns which are usually regular expressions against a chosen message. Several similar patterns increase the score of a message. In contrast, it deducts from the score if any of the patterns did not correspond.Each message's score that reaches a specific threshold is filtered as spam; otherwise it is counted as valid. While some ranking rules do not change over time, other rules require constant updating to be able to cope effectively with the menace of spammers who continuously introduce new spam messages that can easily stay hidden without being noticed from email filters. A good example of a rule based spam filter is SpamAssassin [5].

Previous Likeness Based Spam Filtering Technique: This approach uses memory-based, or instance-based, machine learning methods to classify incoming emails based to their resemblance to stored examples (e.g. training emails). The attributes of the email are used to create a multi-dimensional space vector, which is used to plot new instances as points. The new instances are afterward allocated to the most popular class of its K-closest training instances. This approach uses the k-nearest neighbor (kNN) method for filtering spam emails [5].

Adaptive Spam Filtering Technique: This method detects and filters spam by grouping them into different classes. It divides an email corpus into various groups, each group has an emblematic text. A comparison is made between each incoming email and each group, and a percentage of similarity is computed to decide which probable group the email belongs to [5].

3. PROPOSED METHOD

The general workflow of the proposed method is shown in figure1 and all parts are described as follows:

3.1 Preprosseing

Parser: Raw email data are typically present in Multipart Internet Mail Extension (MIME) format. We utilizes words and hyperlinks present in the body of the email to build a model. Parser consists of the following [4] parts:

MIME Parser: Parses email MIME message and extracts email headers and email body. Email body is further separated into HTML body part and text body part. For emails containing only text MIME part, the parser extracts text and hyperlinks. In a phishing email, these hyperlinks link to the phishing website.

HTML Parser: MIME message-containing HTML body part is included as multipart/html part in the email body section. When the MIME parser detects a HTML part, it invokes the HTML parser to separate text, style-sheets, hyperlinks, and scripts. For the purpose of building model, both text and hyperlinks are considered.

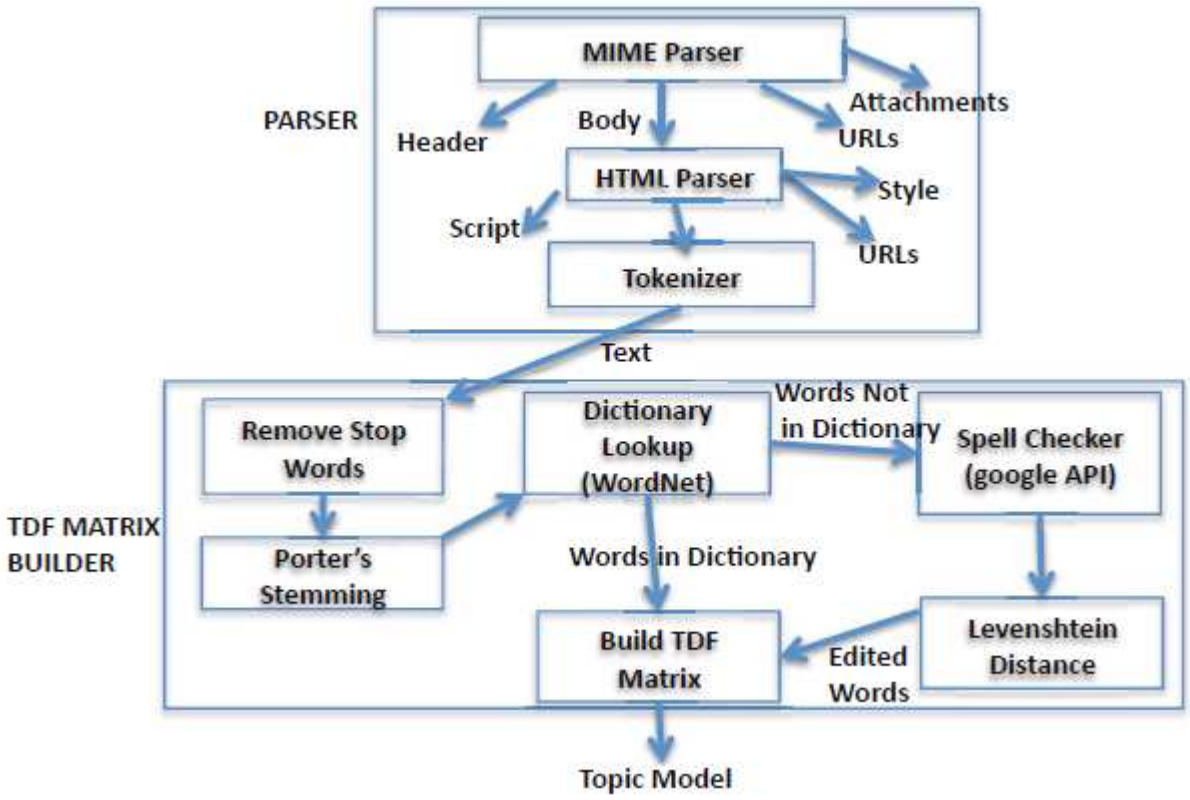

Fig. 1. *Parser and TDF matrix* builder [4]

Tokenizer: This tokenizes email body and hyperlinks into separate words. Tokenizer utilizes white space (tabs, space, new lines) as token delimiters for the text. The hyperlinks are tokenized after replacing all non-alphanumeric characters with space [4].

TDF Matrix Builder: A term-document matrix describes the frequency of terms that occur in a collection of documents. The rows of the matrix correspond to document (d_i) in the collection and the columns correspond to term (w_j) that excist in those documents. For the text part, the terms w_j belong to one of the part-of-speech tags (adjectives, adverbs, nouns, and verbs). For the hyperlinks part, all terms are used to build TDF. The matrix entries $n(d_i, w_j)$ denote the number of times word w_j occurs in document di. Prior to building TDF Matrix, the following preprocessing steps must be accomplished [4].

Stop Words Removal: Stops words are words that do not contain important significance for building the model. Some examples of stop words include the, at, like, etc. We remove stop words from all the tokenized email text [4].

Stemming: Stemming is a method for removing inflexional endings from certain words. For example, the word "consigned", after stemming becomes "consign". Porter's Stemming algorithm is employed to stem words in the email body [4].

Dictionary Lookup: WordNet dictionary is employed to look up words in the dictionary. WorldNet database has a part-of-speech (POS) extractor. It identifies verbs, nouns, adverbs, and adjectives. Words found in the WorldNet database forms the main part of the input for building the TDF matrix. For the hyperlinks TDF, WordNet lookup and spell checker is skipped [4].

Spell Checker: Attackers intentionally misspell words in a phishing email to avoid detection by standard spam filters. For words that are not found in the WordNet database, Google's spell check API is utilized to retrieve words that are similar to the misspelled word [4].

Levenshtein Distance: Levenshtein distance is a metric for measuring the difference between two words. The metric is also called edit distance. It is the minimum edit operations required to transform one word into another. The edit operations include insertion, deletion, and substitution of a new character. In a phishing email, there are misspelled words, which operation is found in the dictionary after edit. Examples include "vulnerability", "youaccounts", etc. Also, there are terms made of garbage characters that are never found in the dictionary. We consider only misspelled words that can be corrected after certain edit operations. After obtaining the suggested words using Google API, Levenshtein distance computed. Only those words whose edit distances are less than a configured threshold (default value of 5) are further used for building the TDF matrix [4].

Build TDF Matrix: For email body text, using words, (specifically adjectives, adverbs, nouns, and verbs), that found directly in the dictionary and edited words using Levenshtein' edit operation, the term-document-frequency matrix is created. For email hyperlinks, all terms used to build a TDF matrix. Thus, we account for misspelled words, conjoined words, and POS tags present in the email body before building the TDF matrix. Once the TDF matrix is built using the components described above, the topic model for phishing detection will be built afterwards.

4. EXPERIMENTAL DESIGN

In this section, we present the details of experiments designed, and the evaluation method.This includes dataset employed, data preparation, training and test strategies, and measures to evaluate performance.

4.1 DataSet

We used email datasets include ham (good) emails and spam (bad) emails from SpamAssassin corpus [8]. SpamAssassin corpus contains a total of 6,047 messages that 4,150 messages of them are good, and the rest is spam. These messages were collected by the SpamAssassin project between years the 2002-2003 and became available to the research community.

4.2 Data preparation

In our combination, there are a total of 10,751 messages, 3,797 phishing, and 6,954 good emails. All the messages were parsed using a MIME parser to separate email headers from the email body. Multipart messages containing HTML parts were further parsed using a HTML parser to extract the body text and hyperlinks. For evaluation, only messages that contain body text and hyperlinks were considered. Thus, messages that failed parser and attachments were not included for building models.

4.3 Training and testing

Experiments were conducted using k-fold cross-validation strategy with a k value of 10. Thus, 90% of the dataset used during training while 10% of that used for testing. In order to build the model, the training data are further split into two parts include 90% for building the topic model and 10% for computing perplexity. Therefore, there were independent datasets for each training, computing perplexity, and testing stage. The TDF matrix builder is then used to build the term-document matrix for each set. Finally, SVM classification applied for detecting spam emails.

4.4 Performance evaluation metrics

The classification performance is measured using the following standard measures such as accuracy, precision, recall, specificity, F-measure, and area under the ROC curve (AUC). They are defined as follows:

Accuracy: proportion of correct predictions to the total Predictions. The value is given by [9]:

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

Positive Predicative (Precision): proportion of predicted positive cases that were correct. The value is defined as[9]:

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

sensitivity (recall) hit rate: This value is defined as:

$$\mathrm{Re}call = \frac{TP}{P} = \frac{TP}{TP + FN} \tag{3}$$

Specificity: proportion of true negatives to the total negatives. The value is defined as[9]:

$$Specificy = \frac{TN}{N} = \frac{TN}{TN + FP} \tag{4}$$

F-measure: This value is defined as:

$$F-measure = 2\frac{\mathrm{Pr}ecision \times \mathrm{Re}call}{\mathrm{Pr}ecision + \mathrm{Re}call} \tag{5}$$

where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. ROC curve is a plot of true positive rate versus false positive rate. The two-dimensional depiction of classifier performance in a ROC curve is reduced to a single scalar value representing expected performance by computing the AUC. The AUC of a classifier is equal to the probability that a classifier will rank a randomly chosen positive example higher than the randomly chosen negative example.

5. RESULTS AND CONCLUSIONS

This section discusses the datasets and evaluation metrics used in experiments, spam detection algorithms, and overall experimental results on pre-processing methods. The contents of the email on the dataset will be preprocessed in advance with one or more preprocessing steps. The preprocessing results are then passed to the spam classifier. This module works to classify whether an email is a spam or not. As can be seen in Figure 2, classifier performances evaluated in the final stage.

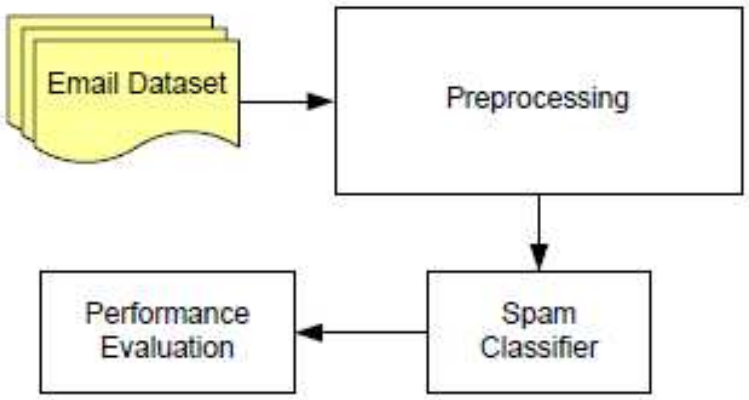

Fig. 2. *Overview of Experiment in Pre-processing for Spam Email Detection*

The dataset used for the experiments is SpamAssassin Corpus [8], a collection of spam and ham emails. This dataset is divided into two parts, training set, and testing set. Results of the pre-processing collection of which is described in section 3 are showed in Figures 3 and 4. For evaluating the performance of the classifier algorithm, measure accuracy, precision, recall, specificity and F-measure are applied. The accuracy of the implemented method is equal to 0.90, the precision is equal to 0.90, the recall is equal to and finally the F-measure is calculated As 0.90. Table1.

Spam Email

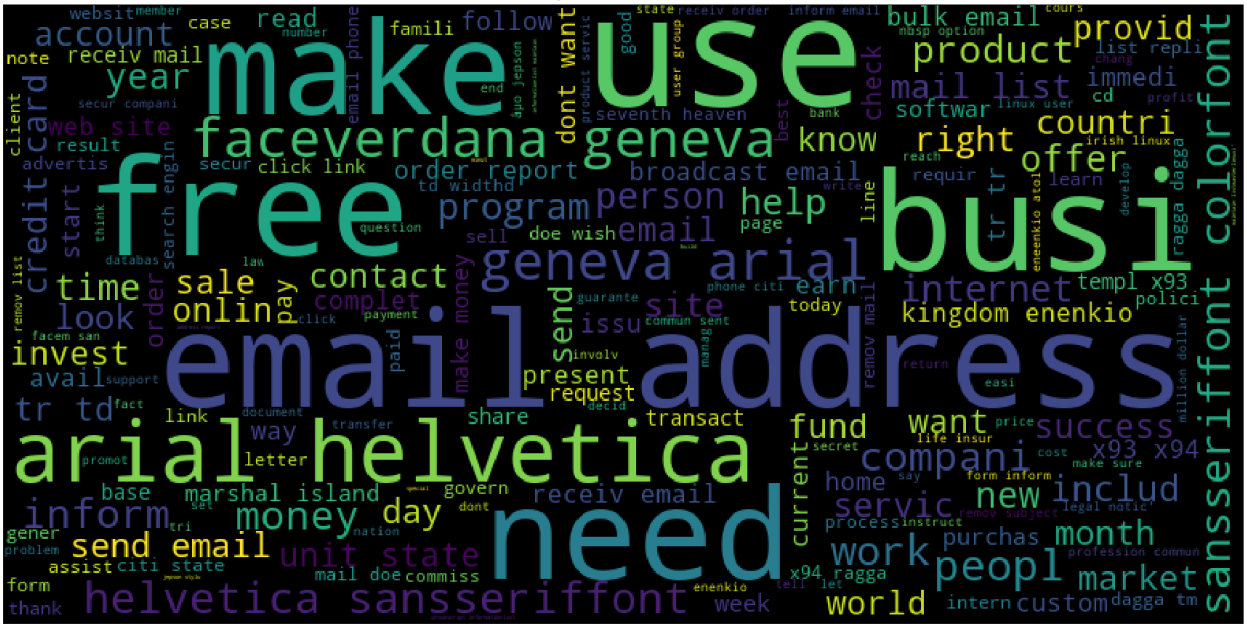

Fig. 3. *Feature of Spam Email*

Non Spam Email

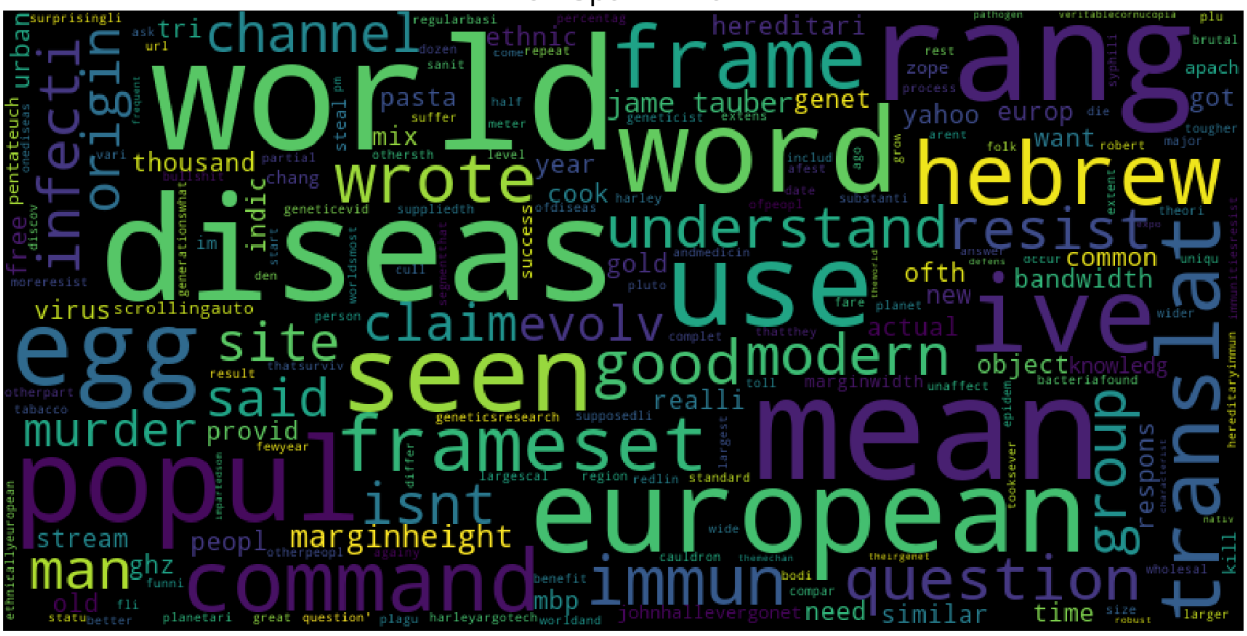

Fig. 4. *Feature of Non Spam Email*

Table 1.*classifiers performance*

Accuracy	Precision	Recall	F-measure
0.90	0.90	0.90	0.90

REFERENCES

[1] S.R. Mirhoseini, B. Minaei (2020). Network Security: Artificial Intelligence method for Attack Detection (Survey Study). 3rd National Conference on Computer, information technology and applications of artificial intelligence, 2020.

[2] Shilin Qiu, Qihe Liu, Shijie Zhou and Chunjiang Wu (2019). Review of Artificial Intelligence Adversarial Attack and Defense Technologies. Available an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license http://creativecommons.org/licenses/by/4.0/, 2019.

[3] Google Says Phishers Stole E-mail From US Officials, Others, PC-World Business Center http://www.pcworld.com/businesscenter/article/229202/google_says_phishers_stole_email_from_us_officials_others.html. Accessed 21 July 2011

[4] Venkatesh Ramanathan and Harry Wechsler (2012). phishGILLNET—phishing detection methodology using probabilistic latent semantic analysis, AdaBoost, and co-training, EURASIP Journal on Information Security 2012, 2012:1.

[5] E.G Dada, J.S Bassi, H Chiroma, S.M Abdulhamid, A.O Adetunmbi, O.E Ajibuwa (2019). Machine learning for email spam filtering: review, approaches and open research problems. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/bync-nd/4.0/). Heliyon 5 (2019) e01802

[6] Abdullah Aljumah, Tariq Ahamad (2016). A Novel Approach for Detecting DDoS using Artificial Neural Networks. In International Journal of Computer Science and Network Security (IJCSNS), 2016.

[7] Mohamed H. Haggag, Ensaf H. Mohammed, Mariam S. El-Rahmany (2017). Social Engineering Attacks Detection Techniques: Survey Study. In International Journal of Engineering and Computer Science (IJECS) Available Online at www.ijecs.in, 2017.

[8] https://spamassassin.apache.org/

[9] S.R. Mirhoseini, M.R. Jahed and M. Pooyan (2016). Improve Accuracy of Early Detection Sudden Cardiac Deaths (SCD) Using Decision Forest and SVM. International Conference on Robotics and Artificial Intelligence (ICRAI2016) Los Angeles, USA, April 20-22, 2016.

Certificate

ISC
Conference code:
99191-95917
Code: IEEEC7-01320124
ID: 132

7th National Congress of News in
Iran Electrical Engineering

Ministry of Science, Research and Technology

This is to certify that

Seyyed Rohollah Mirhoseini, Fatemeh Vahedi, Jalal A Nasiri

gave an oral presentation of their paper entitled:

E-Mail phishing detection using natural language processing and machine learning techniques

at the "7th National Congress of News in
Iran Electrical Engineering"

held on June 13 and 14, 2020 at Alameh Tabatabai University, Tehran, Iran.
This Conference is indexed by ISC and CIVILICA data bases.

Dr. Hosein Pourbashash
Conference Chairman

ieeec.ir
Electrical Engineering

Dr. Mohamadreza Abasifard
Policy Council's Chairman

CIVILICA