ISSN 2347 - 3983

Volume 8. No. 9, September 2020

International Journal of Emerging Trends in Engineering Research

Available Online at http://www.warse.org/IJETER/static/pdf/file/ijeter233892020.pdf

https://doi.org/10.30534/ijeter/2020/233892020

Evaluating the priority of email using machine learning

Do Hoang Long [1], Tisenko Victor Nikolaevich[2],Nguyen The Lam[3], Pham Thi Thuong[4], Nguyen Quang Dam[5]

[1,3,4,5]Information Assurance dept. FPT University, Hanoi, Vietnam, longdhse05220@fpt.edu.vn, lamntse63326@fpt.edu.vn, ThuongPTSE05856@fpt.edu.vn, damnqse05820@fpt.edu.vn,

[2]Department Quality Systems, Peter the Great St. Petersburg Polytechnic University, Russia, St.Petersburg, Polytechnicheskaya, 29, v_tisenko@mail.ru

ABSTRACT

E-mail is an application that is widely used worldwide to help shorten the time, distance, and cost of the mailing process. In this paper, we propose a method of evaluating the role and importance of emails based on the process of analyzing and evaluating the content of emails. Our approach is based on Natural Language Processing and machine learning algorithms to classify emails into important or unimportant. Accordingly, our method is based on 2 main techniques: i) Processing and clustering of keywords in email. For this process, we will use data clustering algorithms. ii) Evaluating email content. For this process, we will use a supervised machine learning algorithm. The research results shown in the paper will provide email systems with a way to identify and classify important emails in order to assist users in sending and receiving emails.

Key words: machine learning, priority of email, Random Forest.

1. INTRODUCTION

Because users often receive many different emails every day, it will be difficult to identify which emails are important and to need to be read and replied early and which emails are just for the track. So we must use the "Priority" concept with email. Accordingly, "Priority" is used to compare two objects or two conditions, where one object/condition must be paid more attention than the other and must be resolved first. The study [1] analyzed and presented problems related to spam email and priority email. Accordingly, the authors have listed and defined a number of emails that are not "Priority" including spam, phishing, spear-phishing, etc. In 2005, at his work publishing, author Jonathan A. Zdziarski [2] stated: Spam email is a large number of unsolicited e-mail messages and most of them are advertising and commercial e-mails. Besides, the paper [3] listed and classified a number of techniques distributing phishing emails. The document [4] listed the level of danger and the impact of email phishing on users. In the study [5], the authors enumerated and classified the methods and techniques for analyzing and detecting spam and phishing emails. To assess the priority of email, documents [1, 6] presented some characteristics and features that need to interest and extract. Accordingly, the features include sender, receiver, time, title, content, and attachments. In this paper, we propose a method of evaluating and classifying the priority of Vietnamese emails based on clustering and classification algorithms. Besides, through some experimental results and evaluations based on user behavior profiles, we have demonstrated that classificating and evaluating the priority of email should be built and ranked based on each user in the email system.

2. RELATED WORKS

Currently, there have been many different studies on the problem of classifying the importance of emails [1]. In which, there are two main methods of evaluating signs based on techniques such as Blacklist, Whitelist, filter by keyword, filter based on social networks, filter SpamAssassin and method based on statistical probability and machine learning. In particular, the problem of applying machine learning algorithms to evaluate and classify email is currently of great interest due to the benefits it brings. Some recent studies on email evaluation and classification are: Chandrasekaran et al. [7] propose a method based on the characteristics of the email structures to detect an email phishing. These characteristics are combined with support vector machine (SVM) algorithm to form a complete email phishing detection system. Toolan and Carthy [8] use the C5.0 algorithm to detect email phishing using 5 features extracted from a dataset consisting of 8,000 emails, half of which are phishing emails and the others are normal emails. Jameel et al. [9] propose a method using neural networks to detect email phishing based on 18 features extracted from email subjects and HTML contents. This method is evaluated with five different structures of the neural networks. In [10], Nizamani apply some classification algorithms such as SVM, Naïve Bayes, J48, and CCM using different sets of features to detect email phishing. Kathsirvalavakumar et al. [11] propose a multilayer neural network structure for email phishing detection. A data preprocessing stage is added to this neural network to reduce the number of input features, and hence reduce the computational cost of the system. Other researches presented in [12, 13] focus on email phishing detection methods based on machine learning algorithms, such as SVM, logistic regression, J48, using 47 features. The experimental results are obtained from Weka toolkit show different accuracy rates corresponding to different selected feature sets.

3. THE MODEL OF EVALUATING THE PRIORITY OF EMAIL BASED ON MACHINE LEARNING

3.1. Model architecture

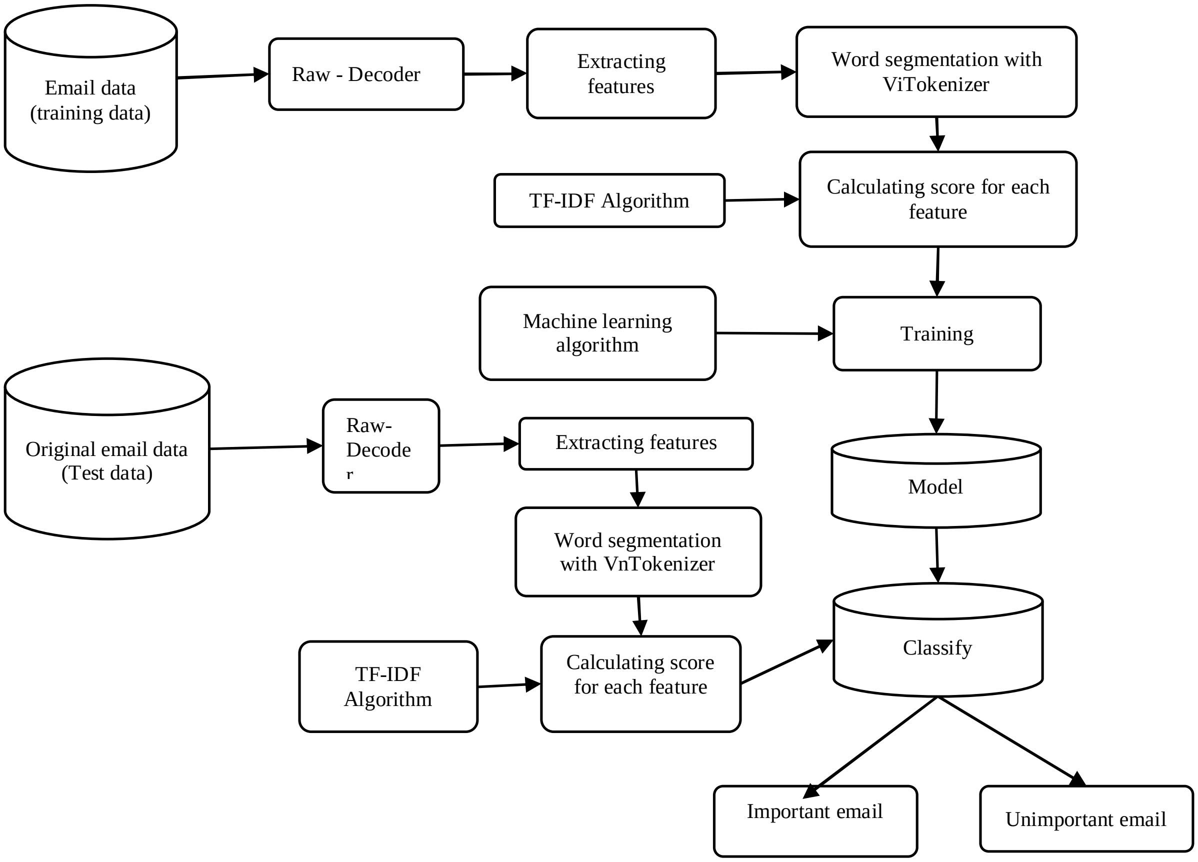


Figure 1: The architecture of the model of evaluating the priority of email

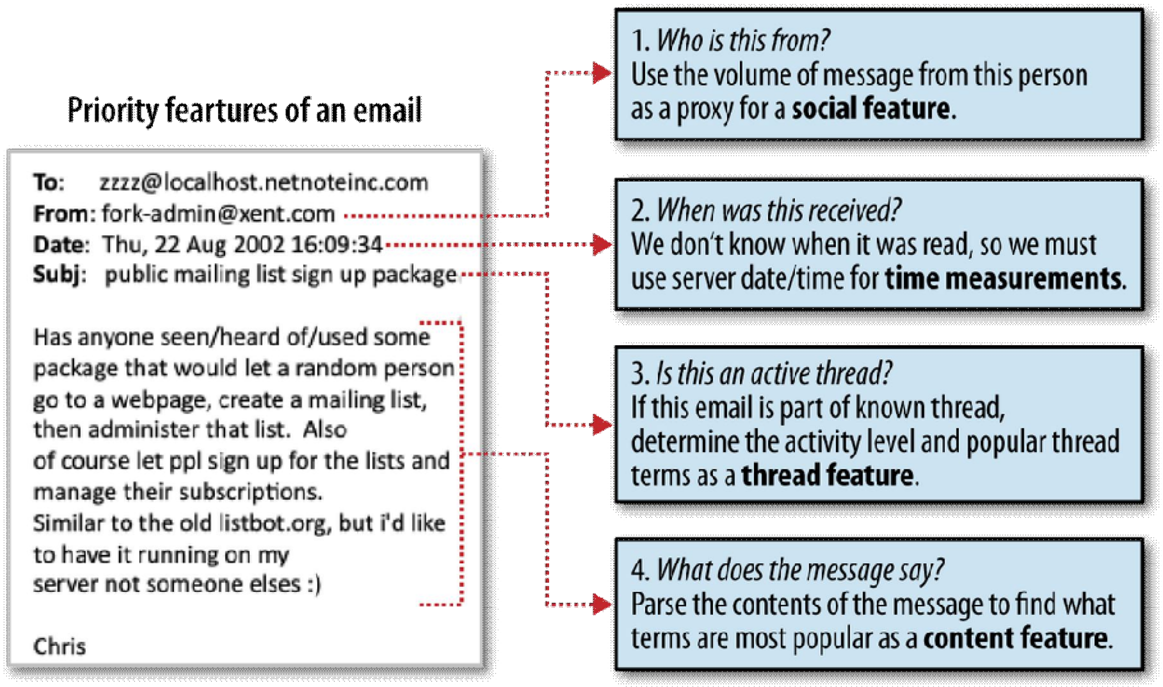
Figure 1 describes the general process of a machine learning process. The process consists of the following 5 steps:

- Entrying data. First, the dataset is uploaded from the file and saved to memory.
- Processing data. At this step, the data uploaded from step 1 will be converted, cleaned and normalized to match the algorithm. The data is converted to be within the same limit, in the same format, etc. Feature extraction and selection process also take place at this step. After that, the data was divided into two sets: 'training set' and 'test set'. The data from the training set is used to build the model. Then the model is evaluated through the test set.
- Training model. Build the model based on the selected algorithm.
- Testing model. The model that was built and trained in step 3 will be tested through the test dataset, and the results generated are used to build a new model. This repetitive process is called “learning” from previous models.
- Deploying model. At this step, the best model is selected for deployment (after a certain number of iterations or when the required result is achieved)

3.2. Feature extraction

3.2.1. The components of the feature

E-mail is a transaction-based medium, social features will be paramount in evaluating the importance of emails [1, 6]. Who was it sent from? Apparently, if a user receives a large volume of emails from a certain address, maybe this user has a strong social connection with the sender. If a user responds regularly to the sender's email address, surely the social connection between the two was strong. So the features worth considering is the sender's address, the receiver's address, and the frequency of responses between them. The important feature that we pay attention to is the time the email is received. Next consider if this email is in a certain email stream or not. Emails in same thread are often on the same subject, and possibly in response to another message. For example in Gmail, it is marked as “RE”. We extract features from the content of the email by using text exploiting techniques. Specifically, if there are common terms in the subject and content of an email that a user receives, future emails containing these terms in the subject and content may be more important than the term that does not appear. This is a common technique and is briefly mentioned in the description about Google Priority Inbox. When considering content features based on both the subject and content of email, there are some terms that are less important in the subject of the email than in the content. Therefore, the relative importance of the common terms in these two features should not be considered equally [1].



Priority feartures of an email

To: zzzz@localhost.netnoteinc.com
From: fork-admin@xent.com
Date: Thu, 22 Aug 2002 16:09:34
Subj: public mailing list sign up package

Has anyone seen/heard of/used some package that would let a random person go to a webpage, create a mailing list, then administer that list. Also of course let ppl sign up for the lists and manage their subscriptions. Similar to the old listbot.org, but i'd like to have it running on my server not someone elses :)

Chris

1. *Who is this from?* Use the volume of message from this person as a proxy for a **social feature**.

2. *When was this received?* We don't know when it was read, so we must use server date/time for **time measurements**.

3. *Is this an active thread?* If this email is part of known thread, determine the activity level and popular thread terms as a **thread feature**.

4. *What does the message say?* Parse the contents of the message to find what terms are most popular as a **content feature**.

Figure 2:. Features that need to be concern [1]

From Figure 2, can see the important features in the email that we use as follows:

a) List of 4 selected features

Table 1: Four important features extracted from the email

No.	**Name**	**Description**
1	Date	The date that email sends
2	From	Email address of the sender
3	Subj	Subject of email
4	Msg	Content of email

b) Weight vectors

Table 2: List of weight vectors

No.	**Name**	**Description**
a	From.weight	The number of occurrences of each email address
b	Senders.df	The number of occurrences of each email address in each subject
c	Thread.weight	$= \frac{\text{The number of occurrences of threa}}{\text{Total conversation time}}$
d	Term.weights	Mean of weights of threads containing these terms
e	Msg.weights	The number of occurrences of each term in all emails

Where:

Counting the number of occurrences of each email address in emails used for training. With the number of occurrences of an email address is x_i, the first weight is $w_1 = \log_{10} x_i$

Filtering emails as response emails. Put the number of occurrences of an email address among the response emails is x_j. The second weight is $w_2 = \log_{10} x_j$

Filtering of threads of emails. Removing threads that do not have replies, calculating the total time of that thread. With thread *i*, put the total time of the thread is *t* with *t* in seconds, the number of occurrence of the thread is *n*. The third weight is $w_3 = \log_{10} \frac{n}{t}$

Using the TF-IDF method, calculating the importance of terms in the content of emails in the sample set. Where *m* is the total terms of the email content, x_j is the importance of each term. The fourth weight is$w_4 = \log_{10} \sum_{m}^{i=1} x_j$

With *n* is the number of terms in the subject of each email, x_j is the importance of each term, the fifth weight is $w_5 = \log_{10} \sum_{n}^{i=1} x_i$[4]

c) How to rank the results

Based on the results from Tables 1 and 2, Table 3 below presents how to rank the result for each email.

Table 3: How to rank the results

Combining	Result	Symbols	Conclusion
2 + a	The rank of this 'from' in all emails	r_1	$r = r_1.r_3.r_3.r_4.r_5$
2 + 3+ b	The rank of this 'from' in this 'subject'	r_2	
3 + c	The rank of this 'subject'	r_3	
3 + d	Mean of ranks of terms in this 'subject'	r_4	

3.3. Classification algorithm

In this paper, we will use a combination of 2 different groups of algorithms. The first group of algorithms relates to preprocessing data. Accordingly, we use 2 algorithms:

- TF-IDF (Term Frequency - Inverse Document Frequency) to calculate the weight to evaluate the importance of a term in an email. A high value represents high importance and it depends on the frequency of the term in the document divided by the frequency of that term in the dataset.
- The K-means algorithm [14] to cluster terms into different groups.

The second group of algorithms is concerned with the classification and evaluation of the importance of the email based on the results of the first group. In this paper, we will experiment and evaluate 3 different classification algorithms including K-Nearest Neighbors (KNN) [14], Random Forest [15], and Logistic Regression [14] algorithms.

4. EXPERIMENTS AND EVALUATIONS

4.1. Collecting and pre-processing data

4.1.1. Collecting data

The dataset used is the dataset collected on the internet. Use Google takeout to retrieve the Mbox file that is mail data of domain name @ fpt.edu.vn. The experimental dataset includes 17 users.Total emails: 61,733. The number of important emails: 20,054.

The collected data is 61,733 emails with max 12 data fields, different languages in the Mbox file format.

4.1.2. Pre-processing data

For each email, we retrieve 4 data fields {'subject', 'from', 'date', 'body'}, filter out emails with other languages and left only Vietnamese emails. Emails saved in Mbox format file are converted to CSV format. With each email, 4 main features {'subject', 'from', 'google_label', 'body'} are extracted. The features will be segmented word and calculated TF-IDF (term frequency–inverse document frequency) score using the following formula:

$$W_{x,y} = tf_{x,y} \text{ x } \log\left(\frac{N}{df_x}\right)$$

Where: tf is the frequency of appearance, df is the number of documents containing term x, N is total number of documents. Figure 3 shows the distribution after word segmentation and scorecard from the TF-IDF score.

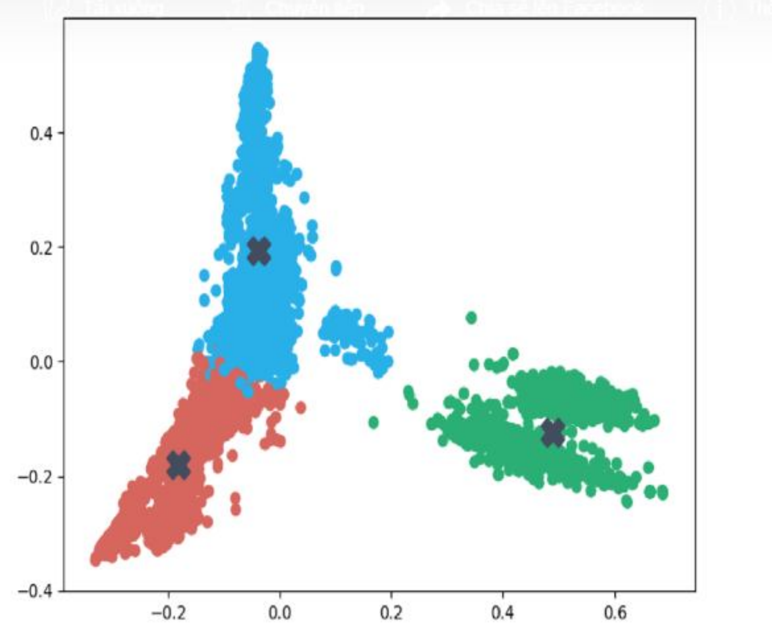


Figure 3: Distribution after word segmentation

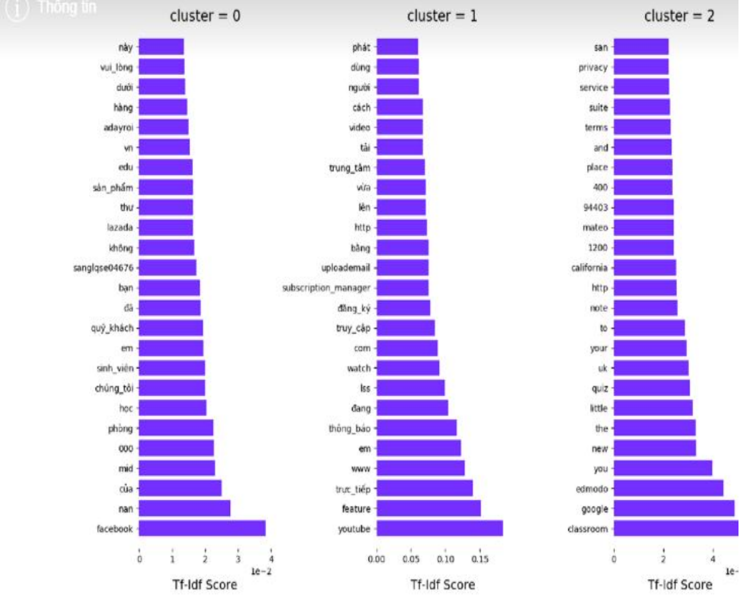


Figure 4: Results of evaluating scores from TF-IDF score using KMeans algorithm with 3 clusters and 100 iterations

4.2. The measures

To evaluate the performance of evaluating the priority of email, 4 different measures are used such as accuracy, precision, recall and f1-score. These metrics are calculated based on the following components:

- True positive (TP) is the number of important emails correctly classified.
- True negative (TN) is the number of unimportant emails correctly classified.
- False positive (FP) is the number of unimportant emails missed classified into important
- False negative (FN) is the number of important emails missed classified into unimportant

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \times 100\ \%$$

$$precision = \frac{TP}{TP + FP} \times 100\ \%$$

$$\mathrm{Re}\ call = \frac{TP}{TP + FN} \times 100\ \%$$

$$F1 - \text{score} = \frac{2 \times precision \times \mathrm{Re}\ call}{precision + \mathrm{Re}\ call}$$

Table 4: Experimental results

User	Model								
	Random Forest			KNN			Logistic Regression		
	AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall
1	0.912	0.892	0.896	0.835	0.862	0.876	0.795	0.84	0.885
2	0.713	0.666	0.670	0.667	0.625	0.632	0.551	0.431	0.571
3	0.953	0.915	0.916	0.846	0.878	0.885	0.715	0.812	0.853
4	0.676	0.617	0.618	0.673	0.637	0.637	0.598	0.551	0.563
5	0.834	0.745	0.745	0.675	0.631	0.631	0.495	0.5	0.515
6	0.838	0.767	0.768	0.683	0.643	0.646	0.67	0.635	0.646
7	0.882	0.841	0.843	0.800	0.794	0.802	0.796	0.782	0.797
8	0.832	0.795	0.802	0.666	0.689	0.705	0.658	0.568	0.694
9	0.849	0.772	0.772	0.722	0.664	0.664	0.65	0.609	0.613
10	0.884	0.795	0.795	0.762	0.702	0.703	0.692	0.644	0.645
11	0.869	0.777	0.776	0.758	0.691	0.692	0.708	0.657	0.659
12	0.949	0.894	0.895	0.862	0.841	0.846	0.778	0.782	0.807
13	0.843	0.775	0.776	0.720	0.673	0.675	0.645	0.606	0.638
14	0.925	0.877	0.879	0.809	0.806	0.815	0.695	0.702	0.767
15	0.808	0.762	0.769	0.684	0.677	0.689	0.631	0.585	0.682
16	0.901	0.819	0.819	0.788	0.718	0.719	0.564	0.431	0.528
17	0.847	0.803	0.809	0.724	0.736	0.754	0.714	0.666	0.753

4.3 Experimental results

From the experimental results in Table 4, can see that the Random Forest algorithm gives better results than Logistic Regression and KNN algorithms. On the other hand, for each different user, there will be different analysis and evaluation results. The cause of this problem is two factors: i) Factors in data. Each user has a different number of important and unimportant emails so if the user has the balance of data, the classification results will be better. ii) Factors in language and habits. Some users have different habits of using words, reading, and replying to emails, so the classifier will give different results if using the same training set. The results of Table 5 below demonstrate this claim.

Table 5: Average efficiency of each algorithm

Model								
Random Forest			KNN			Logistic Regression		
AUC	F1	Recall	AUC	F1	Recall	AUC	F1	Recall
0.854	0.795	0.797	0.746	0.722	0.728	0.668	0.635	0.683

From the above results, we can see that the Random Forest algorithm gives the best classification results among the three algorithms, with the best metrics in turn: AUC: 0.854, F1: 0.795, Recall: 0.797. Obviously, with the dataset including users, the linguistic properties and habits of each user have affected the exact classification of the algorithm

5 . CONCLUSION AND FUTURE DIRECTION

In this paper, based on data clustering and classification algorithms, we have succeeded in analyzing and ranking the importance level of Vietnamese email. The research results of the paper presented in tables 2 and 3 show that our approach is reasonable and correct. Although there are differences in the classification results between users, based on Table 3, we can see the overall results for email classification at an acceptable level. This result shows that if we apply analytics to all emails, a large amount of email data among users is required. Therefore, approaches of analyzing and ranking emails need to build profiles of users in the email system to analyze and evaluate behaviors, habits of each user in order to bring good results. In the future, we will apply the behavioral profile analysis method combined with deep learning algorithms to handle this problem.

REFERENCES

[1] Drew Conway, John Myles White. **Machine Learning for Email: Spam Filtering and Priority Inbox**. *O'Reilly Media*, 2012, ISBN-13: 978-1449314309

[2] Zdziarski, Jonathan. **Ending Spam: Bayesian Content Filtering and the Art of Statistical Language Classification. 1st Edition.** *No Starch Press*, **2005**

[3] APWG. **Phishing Activity Trends Report, 1st 2rd Quarters 2019 . Report APWG**. URL: https://docs.apwg.org/reports/apwg_trends_report_q2_2019.pdf

[4] **Spam and phishing in Q2 2019**. https://securelist.com/spam-and-phishing-in-q2-2019/92379/

[5] Wu, Tingmin; Wen, Sheng; Xiang, Yang; Zhou, Wanlei. **Twitter spam detection: Survey of new approaches and comparative study**. *Computers & Security*. Vol 76, 2018, pp. 265-284.

[6] Douglas Aberdeen, Ondrej Pacovsky, Andrew Slater. **The Learning Behind Gmail Priority Inbox**. https://static.googleusercontent.com/media/research.google.com/vi//pubs/archive/36955.pdf

[7] Chandrasekaran, M., Narayanan, K., & Upadhyaya, 6/2006. **Phishing email detection based on structural properties**,*NYS Cyber Security Conference*. pp. 1-7, 2019.

[8] F. Toolan and J. Carthy. **Phishing detection using classifier ensembles**.*2009 eCrime Researchers Summit, Tacoma, WA, 2009*, pp. 1-9, doi: 10.1109/ECRIME.2009.5342607.

[9] Jameel, Noor Ghazi M., Loay E**. George. Detection of phishing emails using feed forward neural network**, *International Journal of Computer Applications*. Vol 77, pp. 132- 137. 2013.

[10] Nizamani, S., Memon, N., Glasdam, M., & Nguyen, D. D., **Detection of fraudulent emails by employing advanced feature abundance**, *Egyptian Informatics Journal*,Vol 15, pp. 169-174, 2014.

[11] Kathsirvalavakumar, T., Kavitha, K., & Palaniappan, R., **Efficient Harmful Email Identification Using Neural Network**, *British Journal of Mathematics & Computer Science*, Vol 58, pp. 34-41. 2015.

[12] Fang, Yong and Zhang, Cheng and Huang, Cheng and Liu, Liang and Yang, Yue. **Phishing Email Detection Using Improved RCNN Model With Multilevel Vectors and Attention Mechanism**.*IEEE Acsess*. Vol 7. pp. 56329-56340. 2019.

[13] Sa'id Abdullah Al-Saaidah, **Detecting Phishing Emails Using Machine Learning Techniques**, *International Journal of Applied Information Systems*, Vol 12, pp. 12- 16. 2017.

[14] Shai, S.S., Shai B.D.: **Understanding Machine Learning: From Theory to Algorithms**. *Cambridge University Press*. 2014.

[15] Leo, B.: **Random Forests**. *Ma. lear*. Vol 45, No 1, 5-32. 2001.