Voting-based Classification for E-mail Spam Detection

Bashar Al-Shboul[1,*], Heba Hakh[1], Hossam Faris[1], Ibrahim Aljarah[1] & Hamad Alsawalqah[2]

[1]Department of Business Information Technology
[2]Department of Computer Information Systems
The University of Jordan, Queen Rania Al-Abdallah Street, Amman, 11942, Jordan
*E-mail: bashar.shboul@gmail.com

Abstract. The problem of spam e-mail has gained a tremendous amount of attention. Although entities tend to use e-mail spam filter applications to filter out received spam e-mails, marketing companies still tend to send unsolicited e-mails in bulk and users still receive a reasonable amount of spam e-mail despite those filtering applications. This work proposes a new method for classifying e-mails into spam and non-spam. First, several e-mail content features are extracted and then those features are used for classifying each e-mail individually. The classification results of three different classifiers (i.e. Decision Trees, Random Forests and k-Nearest Neighbor) are combined in various voting schemes (i.e. majority vote, average probability, product of probabilities, minimum probability and maximum probability) for making the final decision. To validate our method, two different spam e-mail collections were used.

Keywords: *e-mail spam detection; feature extraction; multi-classifier voting; voting-based classification.*

1 Introduction

Electronic mail (e-mail) is a major communication method that makes use of the Internet. The number of exchanged e-mails is continuously growing [1,2] and along with it the amount of unsolicited e-mails (spam). Spam e-mail appears in different forms and contents (i.e. phishing e-mails, e-mails with offensive, malware and malicious content, advertisement e-mails advertising real or sometimes fake products) [3]. E-mail spam has a serious negative impact on the productivity of e-mail using entities in terms of time, money and network resources. Therefore, e-mail filtering techniques for automatically identifying spam e-mail have been developed. Spam detection and filtering is considered a challenging task with high complexity due to the continuously changing spam patterns (e.g. spammers keep developing new techniques and activities that make the process of identifying or predicting spam much harder [4]) and the large number of features of spam e-mail. Balancing false negatives (i.e. spam e-mails stored in the inbox folder, which annoys the user) and false positives (i.e. good emails transferred to the spam folder, which leads to a loss of valuable

Received September 20th, 2015, Revised November 9th, 2015, Accepted for publication February 19th, 2016.
 ISSN: 2337-5787, DOI: 10.5614/itbj.ict.res.appl.2016.10.1.3

information) is considered a critical process to maintain the maximum level of satisfaction of e-mail service subscribers [4,5].

In this paper we propose an approach for detecting e-mail spam based on a three-stage approach. In the first stage two types of features are extracted from the targeted corpus of emails: e-mail body features and readability features. In the second stage, basic classifiers are trained, based on the extracted features. In this work Decision Trees (J48), Random Forests (RF) and k-Nearest Neighbor (kNN) are used. Finally, the basic classifier results are combined for a final decision (spam or non-spam). Different voting combination schemes were investigated and evaluated: majority vote, average probability, product of probabilities, minimum probability, and maximum probability. The contribution of this work is twofold. First, several body features are proposed and combined with readability features to enhance the spam classification results. Secondly, the investigation and utilization of several voting schemes for e-mail spam detection instead of using only majority vote.

This paper is organized as follows. In Section 2, related works are discussed. In Section 3, details of the selected features are explained. Section 4 shows the experimental setup, while the results are analyzed and discussed in Section 5. Finally, we conclude the findings of this work in Section 6.

2 Related Works

Different data mining techniques and machine learning algorithms have been investigated and developed in the literature for the task of automatically filtering out e-mail spam. In most cases, the previous works focused mainly on one of the following two categories or both.

2.1 Spam Features

This stream of research is aimed at studying the problems of e-mail representation and feature extraction, which may be useful in identifying spam e-mails. Generally, e-mail features are extracted from the text of the body, the subject or the header fields. These types of features are called content-based features. For example in Al-Jarrah, *et al.* [6], the authors proposed a new set of features extracted from the headers of e-emails, which were then used for training common classifiers. In Alqatawna, *et al.* [7], the authors focused on extracting malicious-related features and studied the effect of these features on the effectiveness of different classifiers. In Ruan and Tan [8], the authors proposed various approaches for constructing features for e-mail spam filtering (i.e. a term-frequency analysis approach, a heuristic approach and behavioral-based approaches).

In term-frequency analysis, every word in an e-mail is defined as a feature and a vector of words is used to represent the e-mail. Heuristic approaches mine e-mails in order to discover and generate patterns and rules [8,9]. Behavioral-based approaches construct features based on information related to the behavior of spammers, usually collected from the header, attachment, and/or email flows between groups of e-mail users [8]. In this work, we propose a set of content-based features and combine them with other features from the literature.

2.2 Spam Classifiers

In this stream of research, the performance and evaluation of the classifiers used for identifying e-mail spam receive more attention. The proposed classifiers are trained and evaluated against other common classifiers, some of which are used and applied as a single classifier, for example Support Vector Machines (SVM) [10], Artificial Neural Networks (ANN) [11-13], Naive Bayesian classifiers [14], k-Nearest Neighbor [15] and Decision Trees [16,17]. Generally, there appear to be various challenging problems in the spam-filtering task. Imbalanced class distribution, unequal and uncertain error costs, complex text patterns, the change of spam content with time, and challenges of reactive and adaptive adversaries are a few examples among others [18].

More complex and intelligent classifiers have been proposed to deal with these challenges (e.g. boosting, ensembles and hybrid classifiers) [19-21]. One approach has not been fully explored in e-mail spam filtering, which is to combine classifiers. It has been reported that combining classifiers with different characteristics can improve classification results [22]. In this work, we investigate combining classifiers for the task of e-mail spam filtering.

3 Proposed Method

In order to classify each e-mail, it has to be transformed into a set of features representing spam, shown in Figure 1 as X_1 through X_{35}. Next, various classifiers are selected, shown in Figure 1 as C_1 through C_n. Classifiers are trained a priori to distinguish spam e-mails utilizing a set of training e-mails from freely available e-mail spam collections. After that, a classifier-voting scheme is applied by taking the vote of classification results from the classifiers on the best set of features. Our proposed framework is represented in Figure 1.

3.1 Feature Extraction

In our study, each e-mail was represented by a feature vector consisting of different features.

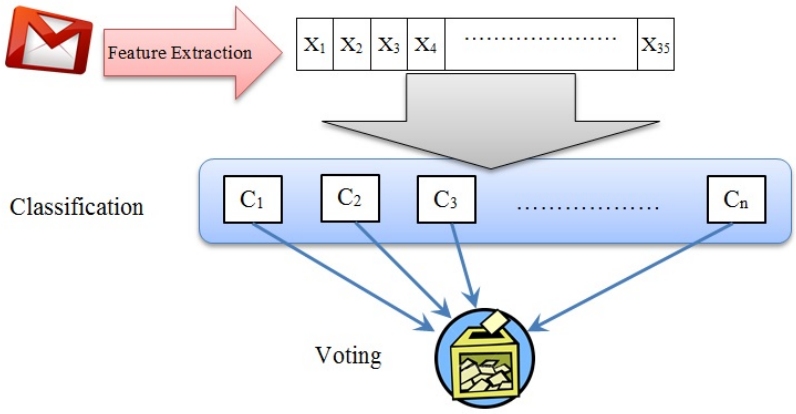

Figure 1 Proposed experimental framework.

In total, 35 features were extracted and then divided into two groups: body and readability features. The details of each group are discussed next.

3.1.1 Body Features

This first group includes the features extracted from the e-mail's body content, excluding the meta-data in the header part. The features, along with their descriptions, are listed in Table 1. As can be noticed, all features are numeric, representing different aspects of spam e-mails. For example, it is uncommon to see a full-length e-mail, consisting of at least 10 lines, written in block letters. Furthermore, many e-mail spammers tend to encode spam words that can be detected through filtering systems with special characters; therefore, their count matters.

To define those features, three undergraduate students were handed 50 e-mails each from which to extract what they believed was a spam indicator. After that, the features with an agreement among at least two were added to our feature list.

3.1.2 Readability Features

The second group illustrates various readability measures, calculating the readability easiness or difficulty of a document, text, or message. Many of these measures have been used previously for spam classification, for example in [23]. To simplify the work, this group was divided into two subgroups: frequency-based readability features and score-based readability features. Frequency-based readability features can be summarized with features X_{22}

through X_{25}, while the remaining features are considered score-based readability features.

Table 1 Body features with their description.

Feature	Description
X_1	E-mail length in words
X_2	Count of duplicate words
X_3	Count of common words (e.g. *is*, *am*, *are*, etc.)
X_4	Average word length
X_5	Minimum word length
X_6	Maximum word length
X_7	Count of uppercase letters
X_8	Count of lowercase letters
X_9	Count of special characters
X_{10}	Longest sequence of adjacent capital letters
X_{11}	Count of spam words[1]
X_{12}	Count of slang words[2]
X_{13}	Count of semicolons
X_{14}	Count of sentences (split by full stops)
X_{15}	Count of alpha-numeric words
X_{16}	Time units
X_{17}	Links (i.e. number of tokens ending with {.net, .com, .jo, etc.}.
X_{18}	Count of emoticons
X_{19}	Count of images/image links
X_{20}	Count of HTML tags
X_{21}	Count of lines

[1] Hubspot: http://blog.hubspot.com/blog/tabid/6307/bid/30684/The-Ultimate-List-of-Email SPAM-Trigger-Words.aspx, last accessed on March 13, 2015.
[2] Internet slang: www.internetslang.com, last accessed on March 18, 2015.

The details of the features are as follows:

1. X_{22} : number of complex words, i.e. words with two or more syllables, one of which is a bound form.
2. X_{23} : number of simple words, i.e. words without affixes or prefixes.
3. X_{24} : document length in number of sentences, i.e. everything that comes after { ? , . , ,} is considered a sentence.
4. X_{25} : average number of syllables per word.
5. X_{26} : Fog Index (FI), the most well-known tool to measure readability. FI has been used in previous works to estimate the years of education required to understand a text on first reading and has shown reliable results. The Fog Index is given as:

$$FI = 0.4 \times (X_{22}/X_{24}) + 100 \times (X_{22}/X_{23}) \quad (1)$$

6. X_{27} : Flesch Reading Ease Score (FRES), used to measure the textual difficulty of the text and given as:

$$FRES= 206.835 \times (X_{23}/X_{24}) – 84.6 \times (X_{25}) \quad (2)$$

7. X_{28} : Flesch-Kincaid Readability Index (FKRI), another Flesch reading test that uses the same equation as (FRES) but with different weighting factors as follows:

$$FKRI = 0.39 \times (X_{23}/X_{24}) + 11.8 \times (X_{25}) – 15.59 \quad (3)$$

8. X_{29} : Simple Measure of Gobbledygook Index (SMOG-I), calculates the difficulty of the writing of a text and given as:

$$SMOG\text{-}I = 1.043\times \sqrt{30 \times \left(\frac{X_{22}}{X_{24}}\right)} + 3.1219 \quad (4)$$

9. X_{30} : SMOG, a variation of SMOG-I in X_{29}, given as:

$$SMOG= \sqrt{30 \times \left(\frac{X_{22}}{X_{24}}\right) + 3} \quad (5)$$

10. X_{31} : FORCAST Index, a test to measure reading skills with the emphasis on the frequency of simple words:

$$FORCAST= 20 - (w/10) \quad (6)$$

where *w* is the number of simple words in a 150-word random sample.

11. X_{32} : Simple Words FI, used to calculate the Fog Index but with the substitution of complex words by simple words:

$$FISimple = 0.4 \times (X_{23}/X_{24}) + 100 \times (X_{22}/X_{23}) -15.59 \quad (7)$$

12. X_{33} : Inverse Fog Index (FI-1)
13. X_{34} : Automated Readability Index (ARI), used to calculate the understandability of a text, given as:

$$ARI = 4.71 (Q/X_{23}) + 0.5 (X_{23}/X_{24}) – 21.43 \quad (8)$$

where Q is the total number of characters read so far.

14. X_{35} : Coleman-Liau Index (CLI), similar to ARI in relying on the characters factor instead of the number of syllables factor. Both are meant to be used in real-time readability measures. The Coleman-Liau Index is given as:

$$CLI = 0.0588 . L – 0.296 . S – 15.8 \quad (9)$$

where L is the average number of characters per word and S is the ratio between sentences and simple words.

3.2 Basic Classifiers

In this work, three state-of-the-art classifiers were used to study their individual behaviors on different feature sets.

1. k-Nearest Neighbor: a simple yet effective classifier. The basic idea is that an entry is classified according to the k closest to other entries. Although closeness can be relative to the method of measurement, the concept still holds. For example, an e-mail can be classified into spam and non-spam based on its similarity to other e-mails sharing the highest number of features. In other words, if e-mail features can be considered to be coordinates in a higher dimensional space, the k-closest e-mails according to some high-dimensional similarity measure should be an indicator for the e-mail class.
2. Decision Trees: a method of organizing features hierarchically according to their importance for class decision-making. Importance can be measured using information gain. Hence, features with high information gain appear at the top of the classification tree. In other words, according to the e-mails in our experiments, each feature's information gain is calculated to build the tree. A new e-mail can thus be classified by traversing through the decision tree.
3. Random Forest: another tree-based classification algorithm. A classification is performed by generating multiple different decision trees, each of which has a different feature structure. After that, a class is assigned based on the majority votes of the different trees.

3.3 Combining Basic Classifiers

In our experiment, different voting strategies were used, namely: product voting, sum voting, max voting, min voting, and majority voting [24]. In product voting, the decision rule combines the a posteriori probabilities generated by the individual classifiers by means of a product rule in order to quantify the likelihood of a hypothesis. It assigns a pattern to the class of which the a posteriori probability product is the maximum. On the other hand, the sum decision rule quantifies the likelihood of a hypothesis by summing the a posteriori probabilities generated by the individual classifiers. It assigns a pattern to the class of which the a posteriori probability sum is the maximum.

Furthermore, the max decision rule quantifies the likelihood of a hypothesis by finding the maximum a posteriori probabilities generated by the individual classifiers by means of a minimum rule. It assigns a pattern to the class of which the maximum a posteriori probability is the maximum. Meanwhile, the min decision rule quantifies the likelihood of a hypothesis by finding the minimum a posteriori probabilities generated by the individual classifiers by

means of a minimum rule. It assigns a pattern to the class of which the minimum a posteriori probability is the maximum. Finally, the majority vote rule quantifies the likelihood of a hypothesis by simply counting the votes received for the hypothesis from the individual classifiers. It assigns a pattern to the class that received the largest number of votes.

4 Experimental Setup

Our proposed method consists of four major steps. First, different sets of features are extracted from each e-mail in the experimental dataset. Then, the feature vector for each e-mail is used to train different classifiers given the best configuration settings for each, as in [23]. After that, we apply the classifiers one by one to the training set and compare the results to rank the classifiers according to their effectiveness. Finally, the best classifiers are selected to vote on each e-mail's class.

For our experiments, two different datasets were used: SpamAssassin and CSDMC2010. SpamAssassin consists of slightly over 8,000 e-mails, among which 500 non-spam e-mails. CSDMC2010 consists of slightly over 4,300 emails, among which approximately 1,400 non-spam e-mails. From each e-mail, 35 features where extracted, generating three groups of feature vectors per e-mail: a body features vector (i.e. contains body features only), readability features vector (i.e. contains readability features only), and a feature vector combining both body and readability features vector. Each group of feature vectors is used solely to train and test each classifier separately. The goal is to see which set of features (i.e. body, readability, and combined body-readability) is the most effective in classifying spam e-mails before proceeding to the next step.

5 Results & Analysis

In this work, among the various available classification algorithms and techniques [25-29], only three different classifiers were utilized and then evaluated i.e. Random Forest (RF), Decision Tree (J48), and k-Nearest Neighbor (IBK). Each of the classifiers was evaluated with different setting combinations in order to optimize the evaluation. The best settings for each classifier according to our experiment are listed in Table 2.

Table 2 Best settings of the utilized classifiers.

Classifier	Parameters
RF	Number of trees: 10
J48	Confidence factor: 0.5, pruning: false
IBK	k = 1

For example, various k values for the (kNN) were tested (i.e. 1, 3, 5, 7, 9), revealing that k = 1 achieved the best effectiveness among others. Meanwhile, different combinations of confidence factor and pruning were tested for J48, revealing the best effectiveness when the confidence factor was set to 0.5 and pruning was disabled.

To increase the ability of correct classification, classifiers were evaluated and assessed according to: accuracy, precision, recall, and hit rate. It is believed that these four evaluation measures can give the closest indication of the best classifier among all others. According to the following confusion matrix, the four evaluation measures are given in Table 3.

Table 3 General Confusion Matrix

		Predicted	
		Non-spam	**Spam**
Actual	**Non-Spam**	tp	fn
	Spam	fp	tn

Accuracy: computes the rate of correctly classified instances of both spam and non-spam, as follows:

$$\text{Accuracy} = (\text{tp} + \text{tn})/(\text{tp} + \text{tn} + \text{fp} + \text{fn}) \quad (10)$$

Precision: computes the proportion of predicted non-spam that was correctly classified, as follows:

$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp}) \quad (11)$$

Recall (true positive rate): computes the rate of predicted non-spam in predicted spam and non-spam, as follows:

$$\text{Recall} = \text{tp}/(\text{tp} + \text{fn}) \quad (12)$$

Hit rate: computes the rate of predicted spam in actual spam and non-spam, as follows:

$$\text{Hit Rate} = \text{tn}/(\text{tn} + \text{fn}) \quad (13)$$

The results reported in Figure 2 (a), (b), (c), and (d) are the accuracy, precision, recall and hit-rate measures of e-mail classification on the SpamAssassin collection, respectively. As can be noticed, readability features didn't achieve high scores in any evaluation measure on their own, when compared to the scores of the body features or the combined body-readability features. Therefore, all further analysis was based on the combined body-readability feature set. In terms of almost all measures, IBK, J48 and RF showed high results in classifying the data correctly. To assist analyzing the results shown in

both Figures 2(b), and 2(c), the F-score[1] measure was calculated using both precision and recall, revealing that IBK topped the other classifiers.

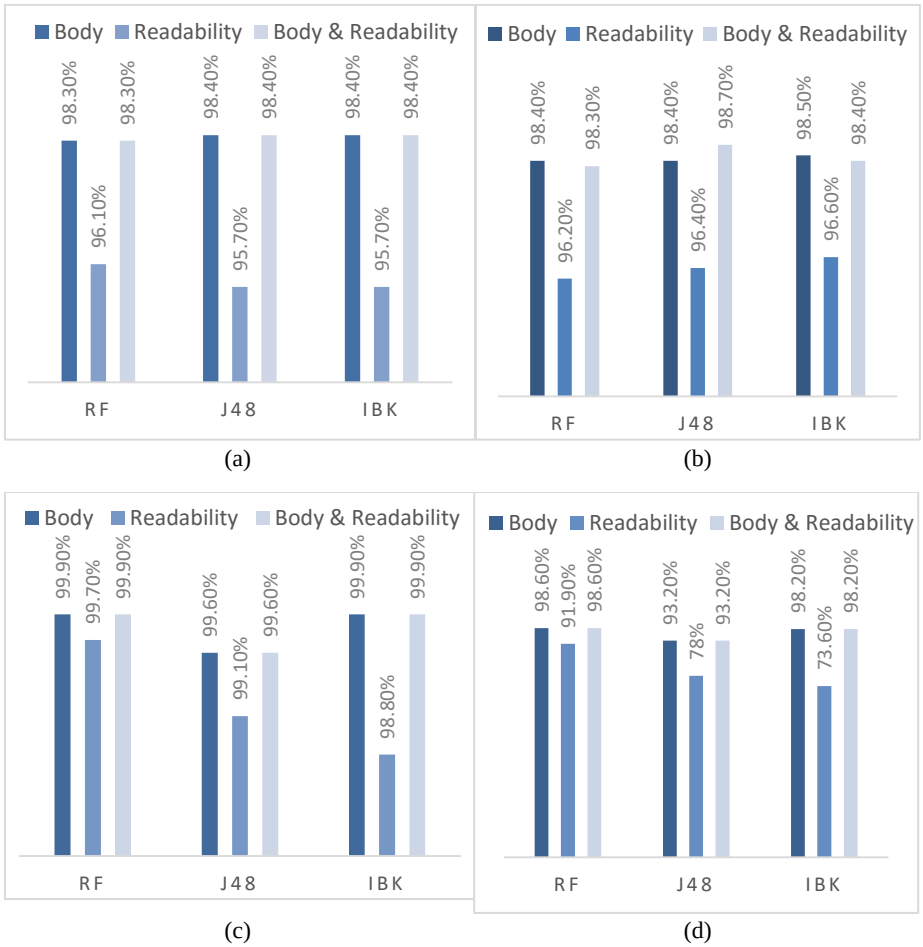

Figure 2 Comparison of evaluation results on the SpamAssassin e-mail collection for various classifiers in terms of (a) Accuracy, (b) Precision, (c) Recall, and (d) Hit Rate.

To test the validity of our results, we applied the same experiment on the CSDMC 2010 e-mail collection as used in [23] for comparison. The results are shown in Table 4, reporting the same three classifiers.

[1] F-score = 2 × Precision × Recall / (Precision + Recall))

As can be noticed, RF had better scores compared to the other two classifiers. Compared to the results of e-mail spam classification on the same collection, as reported in [23], classification using our features showed better scores, given that the RF settings were the same as in [23].

As the individual classifiers' results reported in Figure 2 and Table 4 show slight room for improvement, we decided to consider each e-mail's assigned class for a vote among the three classifiers. The results are reported in Table 5.

Table 4 Precision, hit rate, recall, and accuracy classification results of evaluation on spamassassin & CSDMC 2010 e-mail collections.

	CSDMC2010			**SpamAssassin**		
	J48	**RF**	**IBK**	**J48**	**RF**	**IBK**
Accuracy	95.05%	**96.63%**	93.69%	98.60%	**99.06%**	98.61%
Precision	93.18%	**96.67%**	92.27%	98.90%	**99.03%**	98.64%
Recall	91.15%	**92.60%**	87.52%	99.62%	**99.97%**	99.89%
Hit-Rate	95.90%	**96.61%**	94.30%	93.48%	**99.53%**	98.02%

Table 5 Class voting evaluation measures on both e-mail collections: spamassassin and CSDMC 2010.

	SpamAssassin			**CSDMC2010**		
	Accuracy	**Precision**	**Recall**	**Accuracy**	**Precision**	**Recall**
Majority Vote	98.97%	98.94%	**99.97%**	96.88%	**97.05%**	93.03%
Average Probability	**99.06%**	99.03%	**99.97%**	**96.95%**	**97.05%**	**93.25%**
Product of Probabilities	98.97%	**99.12%**	99.79%	95.41%	94.49%	90.92%
Minimum Probability	98.97%	**99.12%**	99.79%	95.39%	94.42%	90.92%
Maximum Probability	98.96%	**99.12%**	99.77%	95.38%	94.42%	90.86%

As can be noticed, all evaluation measures showed significant improvement over the individual classifier's scores. The only explanation can be that classifiers have different feature preferences; therefore, they don't always agree on the same class. Henceforward, voting assisted the classification measures. It is also noticeable that the average probability voting strategy showed higher scores compared to the other voting strategies; however, we leave studying this phenomenon for future work.

6 Conclusions

In this paper, a new multi-classifier voting-based e-mail spam detection framework is proposed. Several classification algorithms were applied to two given datasets of e-mails for classifying them into spam and non-spam. The results showed that when compared with other works, our features had better

classification power using the same e-mail collections and classification algorithms. Furthermore, the results showed that class-voting assisting the classification showed better effectiveness scores when compared to the results obtained by the individual classifiers.

References

[1] Clark, J., Koprinska, I., & Poon, J., *A Neural Network based Approach to Automated E-mail Classification*, in Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence, p. 702, IEEE Computer Society, 2003.

[2] Kumar, R.K., Poonkuzhali, G., & Sudhakar, P., *Comparative Study on Email Spam Classifier Using Data Mining Techniques*, in Proceedings of the International MultiConference of Engineers and Computer Scientists, **1**, pp. 14-16, 2012.

[3] Cormack, G.V., *Email Spam Filtering: A Systematic Review*, Foundations and Trends in Information Retrieval, **1**(4), pp. 335-455, 2007.

[4] Guzella, T.S., & Caminhas, W.M., *A Review of Machine Learning Approaches to Spam Filtering*, Expert Systems with Applications, **36**(7), pp. 10206-10222, 2009.

[5] Blanzieri, E., & Bryl, A., *A Survey of Learning-based Techniques of Email Spam Filtering*, Artificial Intelligence Review, **29**(1), pp. 63-92, 2008.

[6] Al-Jarrah, O., Khater, I., & Al-Duwairi, B., *Identifying Potentially Useful Email Header Features for Email Spam Filtering*, in The Sixth International Conference on Digital Society (ICDS), 2012.

[7] Alqatawna, J., Faris, H., Jaradat, K., Al-Zewairi, M., & Adwan, O., *Improving Knowledge Based Spam Detection Methods: The Effect of Malicious Related Features in Imbalance Data Distribution*, International Journal of Communications, Network and System Sciences, **8**(05), p. 118, 2015.

[8] Ruan, G., & Tan, Y., *A Three-layer Back-propagation Neural Network for Spam Detection Using Artificial Immune Concentration*, Soft Computing, **14**(2), pp. 139-150, 2010.

[9] Oda, T., & White, T., *Increasing the Accuracy of a Spam-detecting Artificial Immune System*, in Evolutionary Computation, 2003. CEC'03. The 2003 Congress on, **1**, pp. 390-396, IEEE, 2003.

[10] Kołcz, A., & Alspector, J., *Svm-based Filtering of E-mail Spam with Content-specific Misclassification Costs*, in Proceedings of the workshop on text mining (TEXTDM2001), Citeseer, 2001.

[11] Chuan, Z., Xianliang, L., Mengshu, H., & Xu, Z., *A LVQ-based Neural Network Anti-spam Email Approach*, ACM SIGOPS Operating Systems Review, **39**(1), pp. 34-39, 2005.

[12] Androutsopoulos, I., Koutsias, J., Chandrinos, K., Paliouras, G., & Spyropoulos, C., *An Evaluation of Naive Bayesian Anti-spam Filtering*, in Proc. of the Workshop on Machine Learning in the New Information Age, 2000.

[13] Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D. & Stamatopoulos, P., *A Memory-based approach to Anti-spam Filtering for Mailing Lists*, Information Retrieval, **6**(1), pp. 49-73, 2003.

[14] Youn, S., & McLeod, D., *A Comparative Study for Email Classification*, in Advances and Innovations in Systems, Computing Sciences and Software Engineering, pp. 387-391, Springer, Netherlands, 2007.

[15] Lai, C.C. & Tsai, M.C., *An Empirical Performance Comparison of Machine Learning Methods for Spam E-mail Categorization*, in Hybrid Intelligent Systems, 2004. HIS'04. Fourth International Conference on, pp. 44-48, IEEE, 2004.

[16] Fawcett, T., *In Vivo Spam Filtering: a Challenge Problem for KDD*, ACM SIGKDD Explorations Newsletter, **5**(2), pp. 140-148, 2003.

[17] Carreras, X., Marquez, L. & Salgado, J. G., *Boosting Trees for Anti-spam Email Filtering*, in In Proceedings of RANLP-01, 4th International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG, Citeseer, 2001.

[18] Koprinska, I., Poon, J., Clark, J. & Chan, J., *Learning to Classify E-mail*, Information Sciences, **177**(10), pp. 2167-2187, 2007.

[19] Idris, I., Selamat, A. & Omatu, S., *Hybrid Email Spam Detection Model with Negative Selection Algorithm and Differential Evolution*," Engineering Applications of Artificial Intelligence, **28**, pp. 97-110, 2014.

[20] Alexandre, L.A., Campilho, A.C. & Kamel, M., *On Combining Classifiers Using Sum and Product Rules*, Pattern Recognition Letters, **22**(12), pp. 1283-1289, 2001.

[21] Shams, R., & Mercer, R., *Classifying Spam Emails Using Text and Readability Features*, ICDM, pp. 657-666, 2013.

[22] Kittler, J., Hatef, M., Duin, R. & Matas, J., *On Combining Classifiers*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **20**(3), pp. 226-239, 1998

[23] Kotsiantis, S.B., *Supervised Machine Learning: A Review of Classification Techniques*, Informatica, **31**, pp. 249-268, 2007.

[24] Kuhn, M., & Johnson, K., *Applied* Predictive Modeling, Springer-Verlag New York, 2013.

[25] Hand, D., Mannila, H., & Smyth, P., Principles of Data Mining, The MIT Press, Cambridge, MA, USA, 2001.

[26] de Sa, M., *Pattern Recognition Concepts Methods and Applications*, Springer-Verlag Berlin Heidelberg, 2001.

[27] Liu, B., *Web Mining: Exploring Hyperlinks, Contents and Usage Data*, Springer-Verlag Berlin Heidelberg, 2nd ed., 2011.

[28] Faris, H., Aljarah, I. & Alqatawna, J., *Optimizing Feedforward Neural Networks Using Krill Herd Algorithm for E-mail Spam Detection*, Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on, pp. 1-5, 2015.

[29] Rodan, A., Faris, H. & Alqatawna, J., *Optimizing Feedforward Neural Networks Using Biogeography Based Optimization for E-Mail Spam Identification*, International Journal of Communications, Network and System Sciences, Scientific Research Publishing, **9** (1), pp. 19-28, 2016.