Journal of Engineering Science and Technology
Vol. 15, No. 3 (2020) 2090 - 2109

SPAM E-MAIL FILTERING USING POLYNOMIAL NEURAL NETWORKS

MAYY M. AL-TAHRAWI[1,*],
MOSLEH M. ABUALHAJ[2], QUSAI Y. SHAMBOUR[3]

[1] Computer Science Department, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[2] Networks & Information Security Department, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
[3] Software Engineering Department, Faculty of Information Technology, Al-Ahliyya Amman University, Amman, Jordan
* Corresponding author: mtahrawi@ammanu.edu.jo

Abstract

As electronic mails (e-mails) are dominant communication tools, and spammers continue coming up with new spamming techniques to fool spam e-mail filters, the spam e-mail problem is growing rapidly every day; it is considered as a headache for both organizations and computer users, which threatens their security and wastes their time, efforts and resources as well. To overcome this problem, different algorithms have been proposed to filter e-mails and detect the spam ones in the recent decades. In this paper, Polynomial Neural Networks (PNNs) are investigated for the first time in Spam e-mail filtering. PNNs are tested on Ling-spam, the famous benchmark corpus of Spam e-mail. Results of this research work have shown that PNNs is a promising spam e-mail filtering algorithm; PNN filter has recorded above 94% Precision on both Legitimate and Spam e-mails. It has achieved over 99% Recall and around 97% F1-measure in Legitimate e-mails recognition. Regarding Spam e-mails, PNNs has recorded over 99% Accuracy, about 83% F1-mesaure and over 73% Recall.

Keywords: Polynomial neural networks, Spam e-mail filtering.

1. Introduction

This Section starts with an introduction to the spam e-mail problem and some recent statistical facts about the size and effects of this problem. After that, a brief introduction about using machine learning to filter spam e-mails and PNNs is presented.

1.1. Spam e-mail problem

The term “spam” is defined in the Oxford Dictionaries Online [1] as "Irrelevant or unsolicited messages sent over the Internet, typically to large numbers of users, for the purposes of advertising, phishing, spreading malware, etc. ". The first spam e-mail appeared in 1978 by a marketer for digital equipment. He sent the spam message to 400 of the 2600 people on ARPAnet at that time; only few of the message recipients liked the idea. Nowadays, some businesses rely on spam marketing to maximize their revenue, as a spammer can send a large number of e-mail messages instantly paying very little cost. SpamCop [2], a website which helps users in reporting spam, estimated that 60 to 80% of all e-mails are spam. This percentage represents hundreds of millions of spam e-mails every week. On average, more than 5 spam e-mails are sent every second with more than 30 e-mails per second reported as a maximum number.

Spam e-mails usually result in a big waste of valuable resources, including but not limited to, network traffic, computer storage and processor powers, users’ time and effort, threatening security and privacy, offensive content and advertising pornographic content. It also represents a risky tool which can harm the e-mail recipient system, if it has infected attachments. Such problems will result in severe direct or indirect financial losses for both individuals and organizations. According to recent research studies [3], spam costs businesses $20.5 billion every year and future predictions estimate the annual cost to reach from $198 to $257 billion.

1.2. Machine Learning and Spam e-mail filtering

According to a study conducted by Siponen and Stucke [4], spam e-mail filtering is the most common approach of spam e-mail protection in companies worldwide. Spam e-mail filters analyze e-mail content and maybe some additional information, in order to detect spam e-mails. Once a spam e-mail is detected, it can be moved to the spam (junk) folder, deleted or have the sender blocked. Several types of spam e-mail filters are used in the literature, including rule-based filters [5-7] and content-based filters [8-19].

Supervised machine learning algorithms provide an adaptive and automated based approach of spam e-mail filtering which can isolate spam e-mails from legitimate ones efficiently. Given a set of training e-mail messages, which contains e-mails that are labelled as either Spam or Legitimate, a classifier can be taught to discriminate between the two classes of e-mails. More importantly, a machine learning based filter can improve its performance through experience. Probably the first study employing machine learning methods for spam e-mail filtering was published in late 1990s by Sahami et al. [12]. A Bayesian classifier was trained on manually categorized Legitimate and Spam e-mail messages; its performance on unseen cases was remarkable. Since then, several machine learning algorithms have been tested on this task, including boosting decision trees and Support Vector

Machines (SVMs) [9], memory-based algorithms [16], and ensembles of classifiers based on stacking [19].

Spam e-mail recognition is in essence a two-class classification problem. Several automated e-mail classification systems, which are based on Text Categorization (TC) have been developed in the literature [5, 8, 12, 13, 19, 20]. Nevertheless, spam e-mail filtering differs from other text classification tasks as it is a cost-sensitive classification area. Add to this, the spam mails cover a wide spectrum of topics, contain more informal words and sentences as well as more spelling errors.

Polynomial Neural Networks (PNNs) is a supervised machine learning algorithm that was used very early in several applications [21]. In the last decade, PNNs were investigated in both English [22-25] and Arabic [26, 27] TC; they have proved, in these research works, to be competitive to the state-of-the-art text classifiers. In this research, PNNs are investigated in spam e-mail filtering for the first time in the literature. Results of classifying Ling-spam, the benchmark corpus for spam e-mail filtering, using PNNs are very encouraging to conduct further research on using this algorithm in this field.

The following sections of the paper describe, in order, related work on spam e-mail filtering, the data set used and pre-processing operations applied on it, the PNN spam e-mail filtering algorithm, conducted experiments and their settings, results and conclusions.

2. Related Work

Many spam e-mail filters have been developed in the few last decades. Some examples include rule-based filters [5-7], Bayesian approaches [10, 12, 14, 15, 17, 20, 28], k-nearest neighbour (kNN) [14], SVM [9, 11], 18] and Neural Networks (NN) [29-36]. A brief overview of some research work in this field is presented in this section.

Several rule-based classifiers have been tested in e-mail classification [5-7]. Cohen [5] developed a rule-based propositional learning algorithm (RIPPER) to file e-mails into folders. RIPPER was compared against Rocchio in this research and both of them achieved similar accuracies. However, pure rule-based methods have not achieved high performance because spam e-mails cannot easily be covered by rules, and rules do not provide any sense of degree of evidence.

While rule-based approaches make binary decisions, probabilistic techniques provide a degree of confidence of the classification, which is an advantage, especially for cost-sensitive fields like spam e-mail filtering. Bayesian approaches are fast probabilistic classifiers which are suitable to the cost-sensitive spam e-mail filtering application. Several Bayesian approaches were experimented in spam e-mail filtering [10, 12, 14, 15, 17, 20, 28].

In fact, one of the earliest published research works on statistical spam e-mail filtering was conducted by Sahami et al. [12]. The authors experimented a spam e-mail filter based on the Naive Bayes (NB) model. Binary weights were used to represent the attributes selected to build the filter. These attributes have three forms: “words”, “words and phrases” and “words, phrases and non-textual”. Two special corpora were used in this research: The first one consists of 1789 messages with

spam constituting 88.2% of the corpus and the second one has 1183 messages, 18% of them are spam. NB achieved a spam precision varying from 97.1 to 100% and a spam recall from 94.3 to 98.3% on the first corpus, with the best results recorded when using "words, phrases and non-textual" attributes. Experiments on the second corpus recorded a spam precision of 98.9% and recall of 94.2% using "words, phrases and non-textual" attributes. The corpora used in this research are not publicly available, which disables direct comparisons with the results reached in this research.

Androutsopoulos et al. [28] experimented their proposed flexible Bayes model for continuous valued features versus Naïve Bayes. Their proposed model outperformed Naïve Bayes.

Androutsopoulos et al. [14] conducted an evaluation of Naïve Bayes as a spam e-mail filter on their Ling-spam corpus. They studied the effect of the size of the attribute-set, the training set size, using stop-lists and lemmatization on the performance of the NB classifier. The best precision results were recorded using 300 attributes, with lemmatization and stop-words removal applied. They conclude that NB is not viable when deleting blocked messages, which requires additional safety nets to use it in practice as a spam e-mail classifier.

Androutsopoulos et al. [15] tested Naïve Bayes on a collection of personal e-mails, PU1, which the authors made publicly available. The corpus consists of 1099 messages, 43% of which are spam. The authors investigated the effect of training corpus size, attribute set size, lemmatization and stop-word lists. The best results recorded in their experiments were achieved using 50 attributes, stop-lists and lemmatization: 84% for recall, 97% for precision, 91% for weighted accuracy and 4.95 for Total Cost Ratio (TCR).

Pantel et al. [17] implemented a spam e-mail filtering system (SpamCop) which is based on the Naïve Bayesian (NB) approach. The authors compared NB and Ripper [5], a rule-based filter. They used stemming and a dynamically- created list of stop-words. They experimented the use of trigrams rather than words, the effect of the size of the training data size and different ratios of spam and non-spam e-mails. They showed, in their experiments, that SpamCop outperforms Ripper.

Provost's experiments [6] also confirmed that NB outperforms Ripper in terms of classification accuracy on both filing e-mail into folders and spam e-mail filtering.

Rennie [20] used NB in filing e-mail messages into a subset of predefined folders. Stemming was applied, stop words were removed and document frequency (DF) was used for feature selection. Better results were achieved when domain-specific features and hand-crafted phrases were integrated.

Finally, Metsis et al. [10] compared five versions of Naïve Bayes on six new datasets. The five versions of Naïve Bayes are: Multi-variate Bernoulli NB, Multinomial NB with Term Frequency (TF) attributes, Multinomial NB with Boolean attributes, Multi-variate Gauss NB and Flexible Bayes. Multinomial NB with Boolean attributes and Flexible Bayes achieved the best results in their experiments recording an average recall of 97.53 and 95.99 respectively.

Many other machine learning techniques were investigated in spam e-mail filtering. Androutsopoulos [13] found that Naive Bayes and a memory-based technique called TiMBL clearly outperform the keyword-based spam e-mail filter

of Outlook 2000 on the Ling-spam corpora with 16.6% of the messages being spam. A lemmatizer was applied to the corpus, Mutual Information (MI) was used for feature selection and selected terms were represented by binary weights. The number of attributes used in the experiments varied from 50 to 700. The memory-based approach outperformed the Naïve Bayesian filter in this research.

Support Vector Machines (SVM) were also experimented in spam e-mail filtering in many research works [9, 11, 18]. Drucker et al. [9] compared SVM versus Ripper, Rocchio and Boosting decision trees in spam e-mail filtering. The best results were recorded using binary featured SVM and Boosting decision trees. The test sets used in this research are not available for public use. On the other hand, Sculley and Wachman [11] presented Relaxed Online SVM for spam e-mail filtering. The results achieved in their experiments were encouraging though impractical for large datasets, as SVM requires a training time which is quadratic to the number of training e-mails. Finally, Wang et al. [18] investigated using SVM as a spam e-mail filter of the UCI spam corpus implementing a Genetic Algorithm (GA) for feature selection (GA-SVM). Comparing GA-SVM with SVM, better classification results and less number of support vectors are gained using GA-SVM.

Regarding Neural Networks (NNs), little studies were conducted on applying NNs in spam e-mail filtering. This is attributed mainly to the fact that NNs require a lot of time for parameter selection and training. However, previous researches have proved that NNs can achieve very competitive classification accuracy [29-36]. Chen et al. [36] compared four algorithms: NB, decision trees, NN and boosting. NN recorded the best results in their research.

Unfortunately, the direct comparison of the results in these research works is impossible, because of the variations in the corpora, data processing, feature selection and reduction criteria.

3. Datasets and Processing Operations

Research in spam e-mail detection was considerably assisted by publicly available benchmark corpora, like *PU1*, *Ling-spam*, *SpamAssasin* and *Enron* corpus. Different versions of *Ling-spam* were initially developed by Androutsopoulos et al. [14] in 2000 and subsequently many experiments have been reported using these corpora [8, 13-15, 37]. The Bare version of the *Ling-spam* Corpus (i.e., the one with lemmatizer and stop-list disabled) is downloaded from [38] and used in this paper to test the proposed PNN algorithm in spam e-mail filtering. The corpus consists of 2412 Legitimate and 481 spam e-mails distributed in 10 parts corresponding to those used in the 10-fold cross validation experiments; each part consists of Legitimate and spam e-mails, as shown in Table 1.

The following pre-processing steps are applied on the corpus:

i. Each whitespace character (newline, consecutive spaces or tabs) is converted to a single space as whitespace characters have the same effect as single spaces in text classification applications.
ii. All punctuation characters, except underscores and hyphens, are removed and replaced by a single space instead; underscores and hyphens are kept to preserve the meaning of compound phrases, like "object-oriented" for example.

iii. All digits are removed; this is common in text classification applications as this helps in reducing the number of terms and does not affect the accuracy of classification to a great extent.
iv. All non-English letters are removed as they do not help in English text classification applications.
v. Uppercase letters are converted to lowercase correspondence to reduce the number of terms.
vi. The porter Stemmer [39] is used, utilizing our own stop-list with more than 1000 words, so as to reduce noise and number of terms.
vii. Words of length 1, resulting after applying the steps above, are discarded as they do not help in e-mail filtering.

Table 1. The *Ling-spam* Corpus

Part	Number of E-mails	Number of Legitimate E-mails	Number of Spam E-mails
1	289	241	48
2	289	241	48
3	289	241	48
4	289	241	48
5	290	242	48
6	289	241	48
7	289	241	48
8	289	241	48
9	289	241	48
10	291	242	49
Total	**2893**	**2412**	**481**

4. PNNs Filtering Algorithm

Text-based spam e-mail filtering is a text categorization application, in which an e-mail message is classified in one of two classes: Legitimate or Spam. PNNs have been used recently in classifying English [22-25] and Arabic [26, 27] text, and have achieved competitive performance to the state-of-the-art classifiers in this field.

To filter emails using PNNs, each e-mail message is represented as a vector of the weights of the terms selected for building the filter, after the optional application of feature selection and reduction methods. Some common term weighting (representation) methods are binary weights ('1' for the term occurrence in the message and '0' for its absence) [12, 16], term frequency (TF) [28], (term frequency-inverse document frequency (tf-idf) [9] and word-position-based attributes [40, 41]. In this research, two term-weighting schemes are experimented: binary weights and relative frequency weights (term frequency in the e-mail message divided by the length of the e-mail message). The latter one gave superior performance compared to the binary one; thus, only the results of experiments using relative frequency weighting are presented here.

The PNNs model adopted in this research has a 2-layer architecture: the input layer and the output layer. In the input layer, a basis function p(x) of degree 2 is formed of the weights of the terms selected for building the spam e-mail filter. As an example, if the filter is built using two terms t_1 and t_2, each e-mail message is

represented using the weights of these two terms k_1, k_2 respectively. The corresponding quadratic basis function of the message will look as follows:

The output layer then linearly separates the result of the first layer to produce one weight for each of the *2* classes: Legitimate and Spam. The PNN is then trained, using the mean square error criterion, to approximate the required output as follows:

$$p(x) = [\ 1 \quad k_1 \quad k_2 \quad k_1^2 \quad k_1 k_2 \quad k_2^2\ ]^t \tag{1}$$

i. Form a polynomial expansion $\boldsymbol{P_L}$ and $\boldsymbol{P_S}$ of all the basis functions formed in Eq. (1) for the Legitimate and Spam classes respectively as follows:

$$P_L = [\ p(x_{L,1}) \quad p(x_{L,2}) \quad p(x_{L,3}) \ldots \quad p(x_{L,NL})]^t \tag{2}$$

$$P_S = [\ p(x_{S,1}) \quad p(x_{S,2}) \quad p(x_{S,3}) \ldots \quad p(x_{S,NS})]^t \tag{3}$$

where *NL* is the number of Legitimate training e-mails and *NS* is the number of Spam training e-mails in the corpus.

ii. Form a global matrix *P* for the *2* classes as follows:

$$P = [P_L \qquad P_S]^t \tag{4}$$

iii. Find the optimum weights of the *2* classes, that minimize squared error as follows:

$$w_L^{opt} = \underset{w}{argmin} \|Pw - o_L\|_2 \tag{5}$$

$$w_S^{opt} = \underset{w}{argmin} \|Pw - o_S\|_2 \tag{6}$$

where O_L and O_S are the target outputs for the Legitimate and Spam classes respectively.

iv. Apply the normal equations method to compute the classes' weights as follows:

$$P^t P\ w_L^{opt} = P^t\ o_L \tag{7}$$

$$P^t P\ w_S^{opt} = P^t\ o_S \tag{8}$$

v. Finally, compute the two classes' weights as follows:

$$w_L^{opt} = (P^t P)^{-1}\ P^t\ o_L \tag{9}$$

$$w_S^{opt} = (P^t P)^{-1}\ P^t\ o_S \tag{10}$$

To classify a newly coming e-mail as either Legitimate or Spam, follow these steps:

i. Form the basis function *p(x)* for this e-mail.
ii. Assign the new e-mail to the class with the highest score as follows:

$$score = \underset{i}{argmax} w_i^{opt} \bullet p(x) for i = L, S \tag{11}$$

Figure 1 summarizes the architecture of PNNs model.

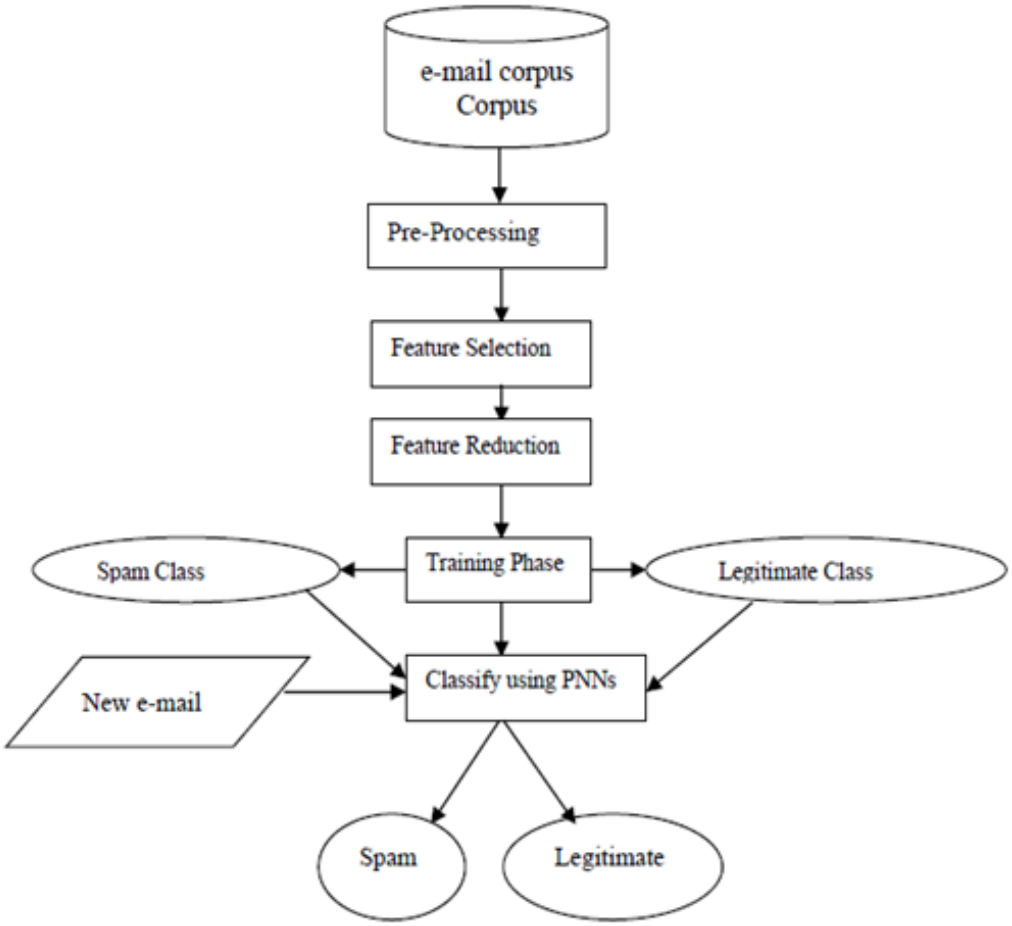

Fig. 1. Architecture of PNNs.

5. Experiments

We start in this section with an explanation of the experimental setups used in this research work, then we explain the performance measures used to evaluate the PNN classifier, and end with presenting, discussing and comparing the results of our experiments versus some other famous machine learning algorithms.

5.1. Experimental setup

Ten-fold Cross Validation is used to test PNNs on Ling-spam in this research; i.e. the classifier is tested 10 times: each time, nine parts are used in training and the remaining part is used in testing. Then, the average performance of the 10-folds is computed. Cross validation gives more confidence in the results, especially with small data sets. PNN algorithm was coded using MATLAB and was executed on a PC with 8 GB RAM and Core i7 CPU for all the experiments conducted in this research.

Previous works have attempted to increase classification accuracy using efficient feature selection methods, such as Information Gain (IG), Mutual Information (MI), Document Frequency (DF), Odds Ratio (OR) and Chi Square

Statistic (CHI). Feature selection aims to reduce the feature space and thus the filtering algorithm resource requirements, get rid of common and noisy terms and work with the most discriminating terms to get accurate filtering results. To reduce memory and time demands of the PNN classifier, Chi Square (CHI) is used for feature selection, and a local-based policy that selects an equal percentage of each class top-scoring CHI terms is used as a feature reduction criterion for building the PNN spam e-mail filter. This combination of feature selection and reduction criteria have led to highly accurate Text Classification (TC) systems [22-27, 42]. I used just 162 terms to build the PNN spam e-mail filter, yet achieved very accurate results, as will be shown in Section 6.2. CHI computes the correlation between each term *t* and each class *c* as follows [43]:

$$\chi^2(\mathrm{t}, \mathrm{c}_L) = \frac{N \times (\mathrm{AD\text{-}CB})^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{12}$$

$$\chi^2(\mathrm{t}, \mathrm{c}_S) = \frac{N \times (\mathrm{AD\text{-}CB})^2}{(A+C) \times (B+D) \times (A+B) \times (C+D)} \tag{13}$$

where: C_L is the Legitimate class, C_S is the Spam class, N is the number of training documents in the corpus, A is the number of documents which belong to the specified class and contain term t, B is the number of documents that belong to the class but do not contain term t, C is the number of documents which do not belong to the class and contain term t and D is the number of documents which neither belong to the class nor contain term t. If a term appears in the 2 classes, its CHI scores are combined in one score using the maximum score.

5.2. Performance measures

The performance of PNNs in classifying the Ling-Spam e-mail corpus is evaluated using the frequently used measures of Precision, Recall, F1-measure, Accuracy, Error Rate and Total Cost Ratio (TCR). Micro-Averaged and Macro-Averaged Precision, Recall and F1-measure are computed as well. Macro-Averaged measures treat spam and legitimate classes equally, even if they have different number of examples in the corpus, while Micro-Averaged measures take into account the number of e-mails in each class.

The misclassification of a Legitimate mail as Spam is much more crucial than the misclassification of a Spam mail as Legitimate. There are different cost approaches directly related with what will be done after the classification. The cost should be chosen high for a filter which deletes detected Spam e-mail messages and can be lower or zero if the filter just marks the e-mails as Spam. In weighted measures, a falsely positive Spam e-mail counts as a λ times more costly mistake than a falsely negative one. Usually, λ values of 1 (the misclassification cost is the same for both error types), 9 (the cost of misclassifying Legitimate messages is increased; notifying senders about blocked messages) and 999 (removing blocked messages) are used to compute weighted performance measures. Weighted Accuracy and weighted TCR are computed to evaluate PNNs in this research work. In all the formulae of the performance measures we use and list below, assume that:

- N_L: the total number of legitimate messages to be classified by the filter.
- N_S: the total number of spam e-mail messages to be classified by the filter.

- $n_{L,L}$: the total number of legitimate messages that were classified correctly as legitimate by the classifier**.**
- $n_{L,S}$: the total number of legitimate messages that were classified erroneously as spam e-mail by the classifier**.**
- $n_{S,S}$: the total number of spam e-mail messages that were classified correctly as spam e-mail by the classifier**.**
- $n_{S,L}$: the total number of spam e-mail messages that were classified erroneously as legitimate by the classifier**.**

5.2.1. Precision

Precision of a model measures how much precise is the model. Precision of the PNN spam e-mail filter is computed as:

$$P_L = \frac{n_{L,L}}{n_{L,L}+n_{S,L}} \tag{14}$$

$$P_S = \frac{n_{S,S}}{n_{S,S}+n_{L,S}} \tag{15}$$

where P_L denotes the Precision of classification of Legitimate messages and P_S denotes the Precision of classification of spam e-mail messages. Precision is a critical measure in Spam e-mail filtering, as the cost of classifying a legitimate email erroneously as Spam can be very high.

5.2.2. Recall

The Recall of the PNN e-mail filter measures how much of the Legitimate or Spam e-mails is classified correctly as Legitimate or Spam respectively. Recall is computed as:

$$R_L = \frac{n_{L,L}}{n_{L,L}+n_{L,S}} \tag{16}$$

$$R_S = \frac{n_{S,S}}{n_{S,S}+n_{S,L}} \tag{17}$$

where R_L denotes the Recall of classification of Legitimate messages and R_S denotes the Recall of classification of Spam e-mail messages.

5.2.3. F1-measure

The F1- measure is the average of Precision and Recall, giving precision and recall even weights. It can be computed as follows:

$$F1_L = \frac{2*P_L*R_L}{P_L+R_L} \tag{18}$$

$$F1_S = \frac{2*P_S*R_S}{P_S+R_S} \tag{19}$$

where $F1_L$ denotes the F_1 measure of classifying Legitimate messages and $F1_S$ denotes the F_1 measure of classifying spam e-mail messages.

5.2.4. Accuracy

Regarding the PNN filter Accuracy, which is the percentage of the correctly classified emails, can be computed as:

$$A = \frac{n_{L,L}+n_{S,S}}{N_L+N_S} \tag{20}$$

while Weighted Accuracy is computed using the following formula:

$$A_w = \frac{\lambda n_{L,L}+n_{S,S}}{\lambda N_L+N_S} \tag{21}$$

5.2.5. Error rate

The Error Rate, which can be defined as the percentage of e-mails that are classified incorrectly, is computed as:

$$Err = \frac{n_{L,S}+n_{S,L}}{N_L+N_S} \tag{22}$$

and the Weighted Error Rate is computed as:

$$Err_w = \frac{\lambda n_{L,S}+n_{S,L}}{\lambda N_L+N_S} \tag{23}$$

5.2.6. Total cost ratio

The Total Cost Ratio (TCR), which measures the time saved by a spam filter, can be computed as:

$$TCR = \frac{N_S}{n_{L,S}+n_{S,L}} \tag{24}$$

and the Weighted Total Cost Ratio is computed as:

$$TCR_w = \frac{N_S}{\lambda n_{L,S}+n_{S,L}} \tag{25}$$

A classifier is considered an efficient e-mail filter if $TCR > 1$ and higher TCR values suggest better performance.

5.3. Results

The results of applying the PNN Spam e-mail filter on the *Ling-spam* corpus using the settings mentioned in the previous sections are summarized in Table 2 through Table 7 and Figs. 2 through 7 below. PNN filter recorded above 94% Precision on both Legitimate and Spam e-mails as is clear in Table 2 and Fig. 2. This highly accurate result is very important in applications like spam e-mail filtering, where the cost of falsely positive Spam is very high. Regarding detecting Legitimate e-mails, PNNs recorded over 99% Recall and around 97% F1-measure as shown in Table 2 and Fig. 2; Recall of Legitimate, and accordingly F1, are higher than the corresponding measures for Spam, since the number of Legitimate training e-mails is higher than the spam ones in the *Ling-Spam* corpus. Spam e-mails detection recorded a Recall over 73% and an F1-mesaure of about 83%. This variation of results is usual in PNN filters which result in better results on classes with more training documents [31].

Regarding spam e-mail filtering Accuracy, PNNs recorded distinguishable results: 0.948453608 for λ =1, 0.986079928 for λ =9 and 0.99168345 for λ =999 as is clear in Table 5 and Fig. 5; These results put PNNs among the topmost performers in Spam e-mail filtering field, as Accuracy is very crucial in Spam e-mail filtering. In fact, PNN filters are known in the literature to record very high classification Accuracies, when combined with class-based CHI feature selection policy [31-36].

Finally, TCR results are 3.266666667 for λ=1, 1.580645161 for λ=9 and 0.024365987 for λ=999 as presented in Table 7 and Fig. 7; these results are consistent with the recorded Accuracy results.

Apparently, all these results put PNNs amongst the top performing Spam e-mail filters on *Ling-spam*, using just a very small subset of the corpus terms.

Table 2. PNN Filter Performance on Legitimate and Spam classes.

Class	Precision	Recall	F1-Measure
Legitimate	94.8617	99.1736	96.9697
Spam	**94.7368**	**73.4694**	**82.7586**

Table 3. Micro-Average PNN performance on SPAM messages.

Micro-Average Precision	94.8454
Micro-Average Recall	94.8454
Micro-Average F1	94.8454

Table 4. Macro-Average PNN performance on SPAM messages.

Macro-Average Precision	**94.7993**
Macro-Average Recall	86.3215
Macro-Average F1	89.8642

Table 5. Weighted Accuracy PNN performance on SPAM messages.

Accuracy (λ =1)	94.8453608
Accuracy (λ =9)	98.6079928
Accuracy (λ =999)	**99.168345**

Table 6. Weighted error rate PNN performance on SPAM.

Error Rate (λ =1)	0.051546392
Error Rate (λ =9)	0.013920072
Error Rate (λ =999)	**0.00831655**

Table 7. Weighted TCR PNN performance on SPAM messages.

TCR (λ =1)	**3.266666667**
TCR (λ =9)	1.580645161
TCR (λ =999)	0.024365987

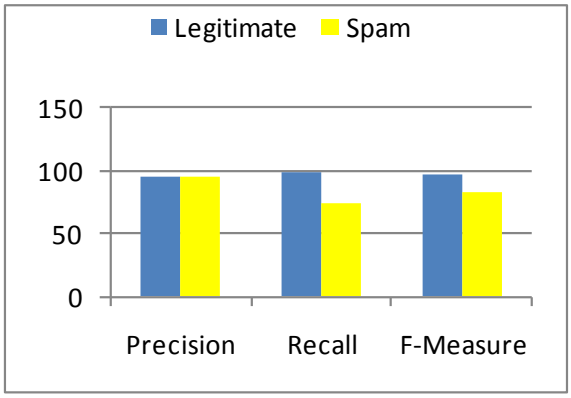

Fig. 2. Legitimate versus spam performance of PNNs on Ling-spam.

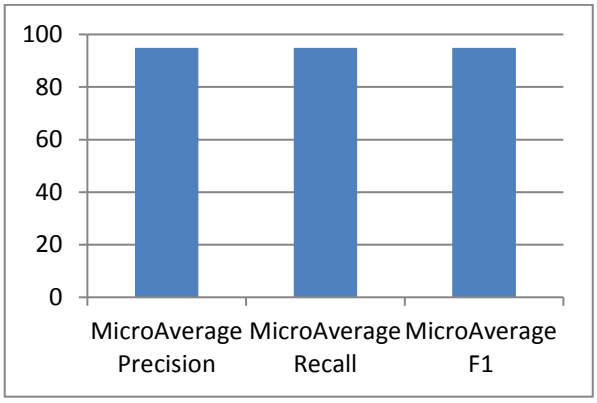

Fig. 3. PNNs Micro Average performance on spam.

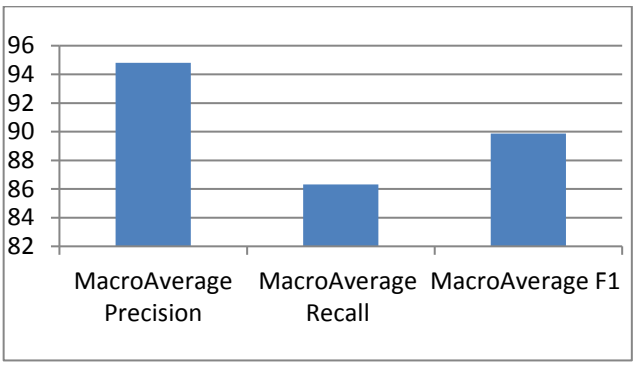

Fig. 4. PNNs Macro Average performance on Spam.

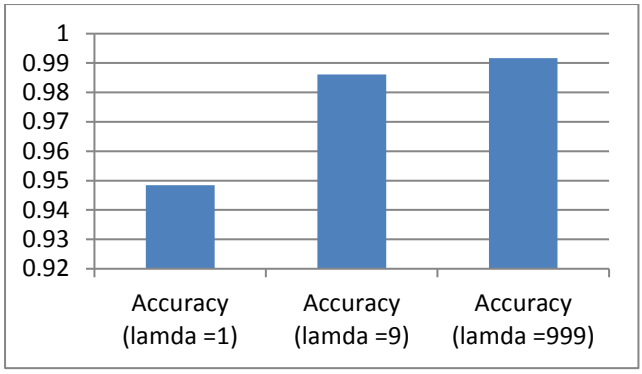

Fig. 5. PNNs Weighted Accuracy performance on Spam.

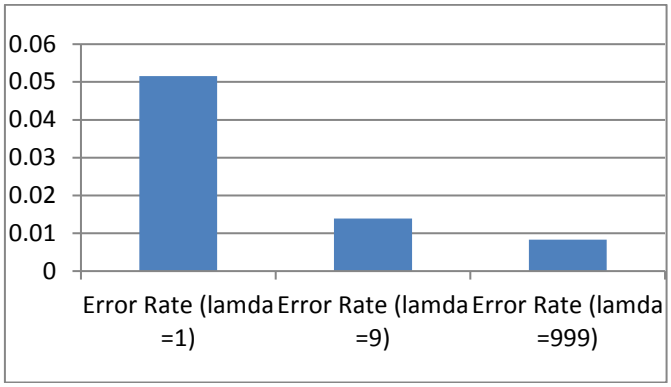

Fig. 6. PNNs Weighted Error Rate performance on Spam.

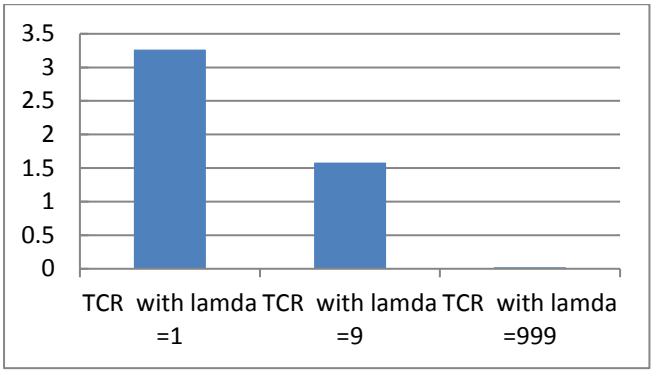

Fig. 7. PNNs Weighted TCR performance on Spam.

5.4. Comparisons with earlier work

Table 8 shows the results of Androutsopoulos et al. [14] research work on the same dataset experimented in this research, using the same number of training and testing Legitimate and spam emails.

Table 8. Results of Androutsopoulos et al. [14] on the Ling-Spam corpus.

Dataset Preprocessing	λ	# Attributes	Spam Recall	Spam Precision	Weighted Accuracy	TCR
Bare	1	50	81.1	96.85	96.408	4.63
Stop-list	1	50	82.35	97.13	96.649	4.96
Lemmatizer	1	100	82.35	99.02	96.926	5.41
Lemmatizer+ Stop-list	1	100	82.78	99.49	97.064	5.66
Bare	9	200	76.94	99.46	99.419	3.73
Stop-list	9	200	76.11	99.47	99.401	3.62
Lemmatizer	9	100	77.57	99.45	99.432	3.82
Lemmatizer+ Stop-list	9	100	78.41	99.47	99.450	3.94
Bare	999	200	73.82	99.43	99.912	0.23
Stop-list	999	200	73.40	99.43	99.912	0.23
Lemmatizer	999	300	63.67	100	99.993	2.86
Lemmatizer+ Stop-list	999	300	63.05	100	99.993	2.86

Comparing PNNs performance results recorded in our research with the results recorded by Androutsopoulos et al. [14] when a lemmatizer and stop-list pre-processing applied, it is clear that PNN filter is a competitive spam filter to *NB*. Furthermore, these competitive results were recorded by PNNs using much less number of attributes (162).

El-Alfy et al. [44] used Group Method of Data Handling (GMDH) based inductive learning approach in detecting spam messages on the Spambase benchmark dataset. They used 82.5% feature reduction, while we worked with less than 1% of the features. Their experiments recorded 91.7% accuracy, which is lower than the accuracy recorded in our experiments.

Ndumiyana et al. [45] used a neural network classifier on a private dataset collected by the authors. Their best recorded spam precision was 93.73%, spam recall was 91.8%, legitimate precision was 91.32% and legitimate recall was 93.75%.

Recently, Abdulhamid et al. [46] compared several algorithms, including multi-layer perceptron, on the Spambase dataset. They did not use feature selection. Their best recorded accuracy was 94.2%, using Rotation Forest algorithm.

Rao et al. [47] constructed four Multilayer Perceptron (MLP) Network models and tested them using 25%, 50%, 75% and 100% of the training UCI benchmark e-mail corpus respectively. Their recorded results in terms of TP rate, FP Rate, Accuracy, Precision, Recall, F-measure and other criteria were reasonable; accuracy recorded using the four models lied between 81.66 and 89.83%. Nevertheless, their recorded accuracy is lower than the one recorded in this research using PNNs.

Finally, Dada et al. [48] conducted a systematic review of some research works on spam e-mail filtering using some common machine learning algorithms using various datasets. Among these research works, the research conducted by Palanisamy et al. [49] tested a hybrid of combined Negative Selection Algorithm

(NSA) and Particle Swarm Optimization (PSO) for spam email classification on the Ling spam dataset. They evaluated their system using only two performance measures: Accuracy and F1. Yet, their recorded classification Accuracy was very low compared with our proposed classifier on the same dataset: 70.48%-83.201% versus 99.168345% using PNNs. Regarding F1 measure, they recorded between 43.546% and 76.85% versus 83% in PNNs.

6. Conclusions

PNNs are investigated, in this research, as a Spam e-mail filter for the first time in the literature of Spam e-mail filtering. PNNs are tested on *Ling-spam*, the commonly used benchmark Spam e-mail corpus. Results of the experiments conducted in this research, together with the comparisons with earlier work reveal that PNNs is a very promising Spam e-mail filter. It recorded either superior or equal performance on Ling-spam to the state of the art spam e-mail filters using a very small but carefully selected subset of features; PNN filter has recorded above 94% Precision on both Legitimate and Spam e-mails. It has achieved over 99% Recall and around 97% F1-measure in Legitimate e-mails recognition. Regarding Spam e-mails, PNNs has recorded over 99% Accuracy, about 83% F1-mesaure and over 73% Recall. Our intended near future work is to compare PNNs directly on the state-of-art Spam e-mail filters using the same dataset, data pre-processing and all experiments settings.

Nomenclatures

A	*Accuracy*
Err	Error Rate
$F1$	is the average of *Precision* and *Recall*
P	Precision of a filter
R	Recall of a filter
S_j	The final score of an email j which determines the class of the email (Spam or Legitimate)

Greek Symbols

λ	Misclassification cost
$\chi^2(t, c_L)$	Correlation between a feature t and a class c_L

Abbreviations

CHI	Chi Square Statistic
GMDH	Group Method of Data Handling
IG	Information Gain
k-NN	k-nearest neighbour
NB	Naive Bayesian
NSA	Negative Selection Algorithm
PNNs	Polynomial Neural Networks
SVM	Support Vector Machines
TC	Text Categorization
TCR	Total Cost Ratio

References

1. Oxford Dictionaries (2018). Retrieved January 1, 2018 from http://www.oxforddictionaries.com/definition/english/spam.
2. Spam Cop (2016). Retrieved February 4, 2016 from https://www.spamcop.net/ces/individuals.shtml.
3. Spam Statistics and Facts (2016). Retrieved January 4, 2016 from http://www.spamlaws.com/spam-stats.html.
4. Siponen, M.; and Stucke, C. (2006). Effective anti-spam strategies in companies: An international study. *Proceedings of the 39th Annual Hawaii International Conference on System Sciences.* Hawaii, USA, 1-10.
5. Cohen, W.W. (1996). Learning rules that classify e-mail. *Proceedings of AAAI spring symposium on machine learning in information access.* California, USA, 18-25.
6. Provost, J. (1999). *Naıve-bayes vs. rule-learning in classification of email*. University of Texas at Austin, Artificial Intelligence Lab. Technical Report AI-TR-99-284.
7. Crawford,E.; Kay, J.; and McCreath, E. (2001). Automatic induction of rules for e-mail classification. *Proceedings of the Sixth Document Computing Symposium.* Coffs Harbour,Australia, 13-20.
8. Carreras, X.; and Marquez, L. (2001). Boosting trees for anti-spam email filtering. *Proceedings of RANLP*. Bulgaria, 58-64.
9. Drucker, H.; Wu, D.; and Vapnik, V.N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks*, 10(5), 1048-1054.
10. Metsis,V.; Androutsopoulos, I.; and Paliouras, G. (2006). Spam filtering with naive bayes-which naive bayes?. *Proceedings CEAS 2006-The Third Conference on Email and Anti-Spam.* Mountain View, California, USA, 1-9.
11. Sculley, D.; and Wachman, G.M. (2007). Relaxed online SVMs for spam filtering. *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*. Amsterdam, the Netherlands, 415-422.
12. Sahami, M.; Dumais, S.; Heckerman, D.; and Horvitz, E. (1998). A Bayesian approach to filtering junk e-mail. *Proceedings of Learning for Text Categorization: Papers from the 1998 workshop*. Madison, Wisconsin, 98-105.
13. Androutsopoulos, I. (2000). Learning to filter spam email: a comparison of a naïve Bayes and a memory based approach. *Proceedings of the Workshop Machine Learning and Textual Information Access, 4th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000).* Lyon, France, 1-13.
14. Androutsopoulos, I.; Koutsias, J.; Chandrinos, K.V.; Paliouras, G.; and Spyropoulos, C.D. (2000). An evaluation of naive bayesian anti-spam filtering. *Workshop on Machine Learning in the New Information Age*. Barcelona, Spain, 9-17.
15. Androutsopoulos, I.; Koutsias, J.; Chandrinos, K.V.; and Spyropoulos, C.D. (2000). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. *Proceedings of the 23rd*

annual international ACM SIGIR conference on Research and development in information retrieval. Athens, Greece, 160-167.

16. Sakkis, G.; Androutsopoulos, I.; Paliouras, G.; Karkaletsis, V.; Spyropoulos, C.D.; and Stamatopoulos, P. (2003). A memory-based approach to anti-spam filtering for mailing lists. *Information retrieval*, 6(1), 49-73.
17. Pantel, P.; and Lin, D. (1998). Spamcop: A spam classification & organization program. *Proceedings of AAAI-98 Workshop on Learning for Text Categorization*. Madison, Wisconsin, 95-98.
18. Wang, H-b.; Yu, Y.; and Liu, Z. (2005). SVM classifier incorporating feature selection using GA for spam detection. *EUC'05 Proceedings of the 2005 international conference on Embedded and Ubiquitous Computing -EUC 2005: Springer*. Nagasaki, Japan, 1147-1154.
19. Sakkis, G.; Androutsopoulos, I.; Paliouras, G.; Karkaletsis, V.; Spyropoulos, C.D.; and Stamatopoulos, P. (2001). Stacking classifiers for anti-spam filtering of e-mail. *Proceedings of the 6th Conference on Empirical Methods in Natural Language Processing (EMNLP 2001), L. Lee and D. Harman (Eds.)*. Carnegie Mellon University, Pittsburgh, PA, USA, 44–50.
20. Rennie, J. (2000). ifile: An application of machine learning to e-mail filtering. *Proceedings of KDD 2000 Workshop on Text Mining*. Boston, MA, USA, 1-6.
21. Assaleh, K.; and Al-Rousan, M. (2005). A new method for Arabic sign language recognition. *EURASIP Journal on Applied Signal Processing, Hindawi Publishing Corporation*, 2136-2145.
22. Al-Tahrawi, M.M.; and Abu Zitar, R. (2008). Polynomial networks versus other techniques in text categorization. *International Journal of Pattern Recognition and Artificial Intelligence*, 22(2), 295-322.
23. Al-Tahrawi, M.M. (2014). The Significance Of Low Frequent Terms in Text Classification. *International Journal of Intelligent Systems*, 29(5), 389-406.
24. Al-Tahrawi, M.M. (2015). Class-Based Aggressive Feature Selection For Polynomial Networks Text Classifiers–An Empirical Study. *UPB Sci Bull, Series C*, 77(2), 1-18.
25. Al-Tahrawi, M.M. (2013). The role of rare terms in enhancing the performance of polynomial networks based text categorization. *Journal of Intelligent Learning Systems and Applications*, 5, 84-89.
26. Al-Tahrawi, M.M.; and Al-Khatib, S.N. (2015). Arabic text classification using Polynomial Networks. *Journal of King Saud University-Computer and Information Sciences*, 27(4), 437- 449.
27. Al-Tahrawi, M.M. (2016). Polynomial Neural Networks Versus Other Arabic Text Classifiers. *Journal Of Software*, 11(4), 418-430.
28. Androutsopoulos, I.; Paliouras, G.; and Michelakis, E. (2004). Learning to filter unsolicited commercial e-mail. *National Center for Scientific Research* " DEMOKRITOS". Technical Report.
29. Bezerra, G.B.; Barra, T.V.; Ferreira, H.M.; Knidel, H.; de Castro, L.N.; and Von Zuben, F.J. (2006). An immunological filter for spam. *Proceedings of the 5th international conference on Artificial Immune Systems CARIS'06: Springer*. Oeiras, Portugal, 446-458.

30. Blanzieri, E.; and Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. *Artificial Intelligence Review*, 29(1), 63-92.
31. Fdez-Riverola, F.; Iglesias, E.L.; Díaz, F.; Méndez, J.R.; and Corchado, J.M. (2007). Applying lazy learning algorithms to tackle concept drift in spam filtering. *Expert Systems with Applications*, 33(1), 36-48.
32. Carpinteiro, O.; Lima, I.; Assis, J.; de Souza, A.; Moreira, E.; and Pinheiro, C. (2006). A neural model in anti-spam systems. *Artificial Neural Networks–ICANN 2006.* Berlin, Heidelberg, 847-855.
33. Wang, B.; Jones, G.J.; and Pan, W. (2006). Using online linear classifiers to filter spam emails. *Pattern analysis and applications*, 9(4), 339-351.
34. Gavrilis, D.; Tsoulos, I.G.; and Dermatas, E. (2006). Neural recognition and genetic features selection for robust detection of e-mail spam. *Proceedings of the 4th Helenic conference on Advances in Artificial Intelligence.* Heraklion, Greece, 498-501.
35. Cui, B.; Mondal, A.; Shen, J.; Cong G.; and Tan, K.L. (2005). On effective e-mail classification via neural networks. *Proceedings of Database and Expert Systems Applications. Lecture Notes in Computer Science, vol 3588:* Springer. Berlin, Heidelberg, 85-94.
36. Chen, D.; Chen, T.; and Ming, H. (2005). Spam Email Filter Using Naive Bayesian, Decision Tree. *Neural Network, and AdaBoost*, 6-30.
37. Vapnik, V. (1995). *The nature of statistical learning theory* (1st edition). USA: Springer Science & Business Media.
38. Androutsopoulos et al. (2000). Ling-Spam data set. Retrieved January, 1, 2015, from http://csmining.org/index.php/ling-spam-datasets.html
39. Porter (1980). Porter Stemmer. Retrieved December 1, 2015, from http://tartarus.org/martin/PorterStemmer/c.txt
40. Vapnik, V.N.; and Vapnik, V. (1998). *Statistical learning theory* (1st edition). New York: Wiley, 156-160.
41. Hovold, J. (2005). Naive Bayes spam filtering using word-position-based attributes and length-sensitive classification thresholds. *Proceedings of the 15th nordic conference of computational linguistics NODALIDA'05.* Finland, 78-87.
42. Al-Tahrawi, M.M. (2015). Arabic Text Categorization Using Logistic Regression. *International Journal of Intelligent Systems Technologies and Applications*, 7(6), 71-78.
43. Zheng, Z.; Wu, X.; and Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, 6(1), 80-89.
44. El-Alfy, E.S.; and Abdel-Aal, M.R.E. (2011). Using GMDH-based networks for improved spam detection and email feature analysis. *Applied Soft Computing*, 11(1), 477-488.
45. Ndumiyana, D.; Magomelo, M.; and Sakala, L. (2013). Spam Detection using a Neural Network Classifier. *Online Journal of Physical and Environmental Science Research*, 2(2), 28-37.

46. Shuaib, M.; Osho, O.; Ismaila, I.; and Alhassan, J. K. (2018). Comparative analysis of classification algorithms for email spam detection. *International Journal of Computer Network and Information Security*, 10(1), 60-67.
47. Rao, A.S.; Avadhani, P.S.; and Chaudhuri, N.B. (2016). A Content-Based Spam E-Mail Filtering Approach Using Multilayer Perceptron Neural Networks. *International Journal of Engineering Trends and Technology (IJETT)*, 41(1), 44-55.
48. Dada, E.G.; Bassi, J.S.; Chiroma, H.; Adetunmbi, A.O.; and Ajibuwa, O.E. (2019). Machine learning for email spam filtering: review, approaches and open research problems. *Heliyon*, 5(6), e01802.
49. Palanisamy, C.; Kumaresan, T.; and Varalakshmi, S.E. (2016). Combined techniques for detecting email spam using negative selection and particle swarm optimization. *International journal of advanced research trends in engineering and technology*, 3(2), 1102-1106.