Polynomial Neural Networks Versus Other Spam Email Filters: An Empirical Study

Mayy Al-Tahrawi [1], Mosleh Abualhaj [2], Sumaya Al-Khatib [3]

[1] *Department of Computer Science, Al-Ahliyya Amman University, Amman, Jordan*
[2] *Department of Networks and Information Security, Al-Ahliyya Amman University, Amman, Jordan*
[3] *Department of Software Engineering, Al-Ahliyya Amman University, Amman, Jordan*

Abstract **– Spam or junk e-mail problems are increasing exponentially due to the huge growth of internet users and their high dependency on e-mails as the main communication means nowadays. Such problems result in huge amounts of time and cost waste for both individuals and organizations. This research paper directly compares the performance of four famous text classification algorithms in classifying e-mails and detecting the spam ones: Polynomial Neural Networks *(PNNs)*, the k-nearest neighbour *(k-NN)*, Support Vector Machines *(SVM)* and Naïve Bayes *(NB)*. Results of the experiments conducted on Ling-spam, the benchmark E-mail corpus, in this research work reveals that *PNNs* is a competitive spam filter to the sate-of-the-art spam filters. It recorded either equal or superior results in most of the performance measures used to evaluate the four spam filters.**

Keywords **– Spam e-mail filtering, Polynomial Neural Networks, k-Nearest Neighbour, Support Vector Machines, Naïve Bayes.**

DOI: 10.18421/TEM91-19
https://dx.doi.org/10.18421/TEM91-19

Corresponding author: Mayy Al-Tahrawi,
Department of Computer Science, Al-Ahliyya Amman University, Amman, Jordan.
Email: mtahrawi@ammanu.edu.jo

Received: 31 October 2019.
Revised: 24 January 2020.
Accepted: 30 January 2020.
Published: 28 February 2020.

1. Introduction

E-mails are very common means of communication among users nowadays, due to their high speed and low cost. Unfortunately, the amount of spam or junk e-mails reaching mailboxes is increasing rapidly every day; a serious problem which threatens users' security and results in a huge waste of time and resources for both individuals and organizations. Many approaches were experimented to resolve this problem; among them was the approach of using text classification algorithms to classify e-mail messages into Spam and Legitimate messages.

In this research, the performance of four famous text classification algorithms in detecting spam e-mails is directly compared on the bench mark e-mail corpus: Ling-Spam, applying the same pre-processing steps on the messages and the same feature selection and reduction criteria for all the algorithms. These algorithms are Polynomial Neural Networks (*PNNs*), the k-Nearest Neighbor (*k-NN*), Support Vector Machines (*SVM*) and Naïve Bayes (*NB*). This paper starts with presenting the Ling-spam dataset in Section 2, followed by an overview of the four classification algorithms tested in this research in Section 3. Section 4 presents the experiments conducted and analyzes the results reached in this paper, while Section 5 presents our conclusions and intended future work.

2. Data Set

The Ling-spam Corpus [1] is used in this research to compare the performance of *PNNs*, *k-NN*, *SVM* and *NB* in spam filtering. Ling-spam consists of 2412 legitimate e-mails and 481 spam e-mail messages. These messages are divided into 10 partitions for use in 10-fold cross validation. Table 1 presents the distribution of the corpus e-mails among the 10 partitions.

Table 1. The ling-spam corpus

Partition	Number of E-mails	Legitimate E-mails	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	□□
7	289	241	48
8	289	241	48
9	289	241	48
10	291	242	49
Totalnumber of e-mails	2893	2412	481

We applied the following pre-processing steps on the corpus messages:

1. Each whitespace character (or sequence of characters) is converted to one space.
2. Punctuation symbols (except hyphens and underscores) are omitted.
3. Digits are omitted.
4. Non-English letters are omitted.
5. Uppercase letters are replaced by their corresponding lowercases.
6. The Porter Stemmer [2] is used, using our special stop-list which consists of over 1000 stop words.
7. Processed words consisting of one character are omitted.

3. The Spam Filtering Algorithms

This section demonstrates an overview of the *PNNs* algorithm, as well as the three filtering algorithms tested against *PNNs* in this research.

3.1. PNNs

Polynomial Neural Networks (*PNNs*) have been used very early for classification [3], [4], [5] as well as sign language recognition [6]. Later on, PNNs have proved to be efficient classifiers of both English texts [7] and Arabic texts [8].

The architecture of the *PNN* model has two layers. In the first layer, known as the input layer, the set of N features x(x_1, x_2, ..., x_N), are used to form a set of monomial basis functions p(x) of degree two. One basis function p(x) is formed for each email as follows [4]:

$$\prod_{j=1}^{N} x_j^{k_j} \text{ , where } k_j \geq 0 \text{ and } 0 \leq \sum_{j=1}^{N} k_j \leq K \quad (1)$$

for each input vectors x_i and each class j. The final score is derived by computing the average total score of all vectors and this final score determines the class of the email (spam or legitimate):

$$S_j = \frac{1}{M} \sum_{i=1}^{M} w^t \bullet p(x_i) \quad (2)$$

For more explanation of the algorithm, the reader can refer to [7].

3.2. K-Nearest Neighbour (k-NN)

k-Nearest Neighbour (*k-NN*) classification algorithm is a famous statistical algorithm which has been used very early in Text Categorization (*TC*). It is a simple algorithm which classifies a new unseen text by finding the *k* nearest neighbours of this text, and using majority voting of these neighbours to determine the class of this text. Cosine similarity between documents vectors is used as a basis of classification.

3.3. Support Vector Machines (SVM)

Support Vector Machines (*SVM*) is a famous supervised machine learning method that was introduced by Vapnik [9] and [10]. *SVMs* was heavily investigated in classification tasks including *TC* [11] and spam e-mail filtering [12] and [13]. In brief, *SVM* creates a hyperplane that separates two or more classes by giving weights to the feature vectors which are used to maximize the margin between the hyperplane and the closest points of data. For more details of the algorithm, the reader can refer to [11].

3.4. Naïve bayes (NB)

Naïve Bayes(*NB*) is a famous probabilistic classifier which was used widely in both *TC* [14], [15], [16] and Spam Filtering [17], [18] and [19]. To classify an email or a document, *NB* depends on computing the joint probabilities of terms and classes, assuming term independence.

In*NB*, the probability of a class *C*, given an email *ej*is computed as follows [20]:

$$P(C \mid e_j = \frac{P(e_j \mid C)P(C)}{P(e_j)} \quad (3)$$

4. Experiments and Results

In this research, 10-fold Cross Validation is used to test *PNN, k-NN, SVM* and *NB* on Ling-spam. That is, each classifier is tested 10 times; each time, nine parts are used in training and the tenth part is used in testing. At the end, the average performance of the 10 rounds is computed. This approach gives more reliable results when working with small datasets.

4.1. Feature Selection

Feature selection is commonly used in classification to provide more accurate and efficient classifiers, as it reduces noise and decreases the resources needed for classification. Some well-known feature selection methods are *Information Gain (IG), Document Frequency (DF)* and *Chi Square Statistic (CHI).*

In this research, Chi Square (*CHI*) is used for feature selection. Then, an equal percent of the topmost discriminating features of each class is selected for building the four filters. This approach has resulted in high accuracy Text Classification (*TC*) systems [7] and [8]. The resulting features after reduction are just 162 features; these features are used to build all the spam filters used in this research. Despite this very low number of features, *PNN* filter was able to achieve excellent results, as will be shown in Section 4.3.

CHI computes the correlation between each feature *t* and each of the two classes C_L and C_S as follows [14]:

$$\chi^2(t,c_L)=\frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \tag{4}$$

$$\chi^2(t,c_S)=\frac{N\times(AD-CB)^2}{(A+C)\times(B+D)\times(A+B)\times(C+D)} \tag{5}$$

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 a certain class and contain feature *t*, *B* is the number of documents that belong to the class but do not contain feature *t*, *C* is the number of documents which do not belong to the class and contain feature *t* and *D* is the number of documents which neither belong to the class nor contain feature *t*. If a feature appears in more than one class, its maximum *CHI* score is adopted.

4.2. Evaluation Measures

This section provides an explanation of the performance measures used to evaluate the spam filters used in this research. The performance of *PNN*, *k-NN*, *SVM* and *NB* in classifying the Ling-Spam e-mail corpus is evaluated using Precision, Recall, F1-measure, Accuracy, Error Rate and Total Cost Ratio (*TCR*). In addition, MicroAveraged and MacroAveraged Precision, Recall and *F1*-measure are also computed to give an idea of the overall performance of the spam filters.

Classifying a legitimate mail erroneously as spam is much costlier than the opposite case. Consequently, several cost approaches are commonly used in spam filtering research work; the cost is set to high for a filter which deletes detected spam messages and to low or zero if the filter only marks the e-mails as spam. In weighted measures, a falsely positive spam counts as a λ times costlier mistake than a falsely negative one. Usually, λ values of 1 (the misclassification cost is the same for both error types), 9 (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 the four filters tested in this research work. In all the formulae of the performance measures we use and list below, assume that:

- N_L represents the total number of legitimate messages to be classified by the filter.
- N_S represents the total number of spam messages to be classified by the filter.
- $n_{L,L}$ represents the total number of legitimate messages that were classified correctly as legitimate by the classifier.
- $n_{L,S}$ represents the total number of legitimate messages that were classified erroneously as spam by the classifier.
- $n_{S,S}$ represents the total number of spam messages that were classified correctly as spam by the classifier.
- $n_{S,L}$ represents the total number of spam messages that were classified erroneously as legitimate by the classifier.

4.2.1. Precision

Precision of a spam filter is computed as:

$$\boldsymbol{P_L} = \frac{n_{L,L}}{n_{L,L}+n_{S,L}} \tag{6}$$

$$\boldsymbol{P_S} = \frac{n_{S,S}}{n_{S,S}+n_{L,S}} \tag{7}$$

where P_L refers to the Legitimate classification Precision and P_S refers to the Spam classification Precision.

4.2.2. Recall

The Recall of the filter is computed as:

$$\boldsymbol{R_L} = \frac{n_{L,L}}{n_{L,L}+n_{L,S}} \tag{8}$$

$$\boldsymbol{R_S} = \frac{n_{S,S}}{n_{S,S}+n_{S,L}} \tag{9}$$

where R_L refers the Legitimateclassification *Recall* and R_S refers to the Spamclassification *Recall.*

4.2.3. FI-measure

The *F1-measure* is the harmonic average of *Precision* and *Recall*. It can be computed as follows:

$$\boldsymbol{F1_L} = \frac{\boldsymbol{2*P_L*R_L}}{\boldsymbol{P_L+R_L}} \quad (10)$$

$$\boldsymbol{F1_S} = \frac{\boldsymbol{2*P_S*R_S}}{\boldsymbol{P_S+R_S}} \quad (11)$$

where $F1_L$ is the *F1-measure* for Legitimate e-mails classification and $F1_S$ is the *F1-measure* for Spam e-mails classification.

4.2.4. Accuracy

The Filter Accuracy, it is computed as:

$$\boldsymbol{A} = \frac{\boldsymbol{n_{L,L}+n_{S,S}}}{\boldsymbol{N_L+N_S}} \quad (12)$$

and the Weighted Accuracy is computed as:

$$\boldsymbol{A_w} = \frac{\boldsymbol{\lambda n_{L,L}+n_{S,S}}}{\boldsymbol{\lambda N_L+N_S}} \quad (13)$$

4.2.5. Error rate

The Error Rate is computed as:

$$Err = \frac{n_{L,S}+n_{S,L}}{N_L+N_S} \quad (14)$$

while the Weighted Error Rate is computed as:

$$Err_w = \frac{\lambda n_{L,S}+n_{S,L}}{\lambda N_L+N_S} \quad (15)$$

4.2.6. Total cost ratio (TCR)

Total Cost Ratio (TCR) is computed as:

$$TCR = \frac{\boldsymbol{N_S}}{\boldsymbol{n_{L,S}+n_{S,L}}} \quad (16)$$

while the Weighted Total Cost Ratio is computed as:

$$\boldsymbol{TCR_w} = \frac{\boldsymbol{N_S}}{\boldsymbol{\lambda n_{L,S}+n_{S,L}}} \quad (17)$$

Efficient filters have a *TCR* value higher than 1, with higher *TCR* values suggesting better performance.

4.3. Results

Results of applying *PNN*, *k-NN*, *SVM* and *NB* Spam filters on the Ling-spam corpus using the settings mentioned in the previous sections are summarized in Tables 2.- Table 8. below.

Table 2. Spam filters performance on legitimate and spam classes

	PNNs			SVM			NB			k-NN		
Class	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure	Precision	Recall	F1-Measure
Legitimate	94.8617	99.1736	96.9697	92.3664	100	96.0317	93.0502	99.5868	96.2076	93.0233	99.1736	96
Spam	94.7368	73.4694	82.7586	100	59.1837	74.359	96.875	63.265306	76.5432099	93.9394	63.2653	75.6098

Table 3. MicroAverage performance on SPAM messages

Algorithm	PNNs	SVM	NB	k-NN
MicroAverage Precision	94.8454	93.1271	93.4708	93.1271
MicroAverage Recall	94.8454	93.1271	93.4708	93.1271
MicroAverage F1	94.8454	93.1271	93.4708	93.1271

Table 4. MacroAverage performance on SPAM messages

	PNNs	SVM	NB	k-NN
MacroAverage Precision	94.7993	96.1832	94.9626	93.4813
MacroAverage Recall	86.3215	79.5918	81.426	81.2194
MacroAverage F1	89.8642	85.1954	86.3754	85.8049

Table 5. Weighted Accuracy performance on SPAM message.

	PNNs	SVM	NB	k-NN
Accuracy (λ =1)	94.8453608	93.1271478	93.4707904	93.1271478
Accuracy (λ =9)	98.6079928	99.1019308	98.7876066	81.2194
Accuracy (λ =999)	99.168345	99.9917289	99.5794166	99.1662772

Table 6. Weighted Error Rate performance on SPAM messages

	PNNs	SVM	NB	k-NN
Error Rate (λ =1)	0.051546392	0.068728522	0.065292096	0.068728522
Error Rate (λ =9)	0.013920072	0.008980692	0.012123934	0.016165245
Error Rate (λ =999)	0.00831655	0.0000827106	0.004205834	0.008337228

Table 7. Weighted TCR performance on SPAM messages

	PNNs	SVM	NB	k-NN
Error Rate (λ =1)	0.051546392	0.068728522	0.065292096	0.068728522
Error Rate (λ =9)	0.013920072	0.008980692	0.012123934	0.016165245
Error Rate (λ =999)	0.00831655	0.0000827106	0.004205834	0.008337228

Table 8 FP and FN Rate performance on SPAM messages

	PNN	SVM	NB	K-NN
FP Rate	0.008264	0	0.004132231	0.008264463
FN Rate	0.265306	0.408163265	0.367346939	0.367346939

4.4. Analysis of Results

It is obvious from the results presented in Table 2-Table 8 above, as well as Fig.1 –Fig.16 below that *PNNs* filter is a competitive spam filter to the state-of-the art algorithms experimented in this research. As an overall performance on both Legitimate and Spam e-mails, it is clear from Table 2 and Figs. 2 and 6 that *PNN* was the winner algorithm among the four tested algorithms.

PNNs filter recorded more than 94% Precision on Legitimate and Spam e-mails. Regarding detecting Legitimate e-mails, *PNNs* outperformed *SVM, k-NN* and *NB* in the overall *F1*-measure as is clear from Table 2 and Fig. 13; *PNNs* achieved more than 99%

Recall and approximately 97% *F1*-measure on legitimate messages.

Regarding spam e-mails detection, *PNNs* recorded a Recall of more than 73% and an *F1*-mesaure of approximately 83% outperforming *SVM, k-NN* and *NB* as is clear from Table 2. and Figs. 15. and 16. As an overall performance of Precision and Recall for the four filters, *PNNs* recorded the highest Micoaverage-F1 and Macroaverage-F1measure on both Legitimate and Spam classes, followed by *NB*, and finally comes *k-NN* and *SVM* as presented in Figs. 2 and 5. Nevertheless, *SVM* recorded 100% Precision on Spam e-mails and 100% Recall on Legitimate e-mails.

As is clear from Table 5. and Fig. 6., regarding Spam filtering Weighted Accuracy, PNNs outperformed the other three filters for λ =1, SVM was the winner for λ = 9 and all the filters recorded above 99% Accuracy for λ = 999, with PNNs recording 0.986079928 Accuracy for λ =9 and 0.99168345 for λ =999. In fact, PNN filters are known in the literature to record very high classification Accuracies, when combined with class-based CHI feature selection policy [7] and [8].

Regarding Weighted Error Rate, the four filters recorded their best results for λ=999, with SVM as the best performer, followed by NB, then follows PNNs and finally k-NN. These results are presented in Table 6. and Fig. 7.

As for *TCR*, all the filters recorded their best values for λ =1, with *PNNs*, *k-NN* and *SVM* recording the best (equal) *TCR* results outperforming *NB* to a great extent; these results are consistent with the recorded Accuracy results. *TCR* results are summarized in Table 7. and Fig. 8.

Finally, regarding Spam *FN* rate, *PNNs* outperformed the three filters to a great extent as shown in Table 8. and Fig. 10. In fact, *FN* Rate is very critical in evaluating a filter's performance due to its dangerous effect on users and resources.

Apparently, all these results reveal that *PNNs* is one of the best Spam e-mail filtering algorithms on Ling-spam.

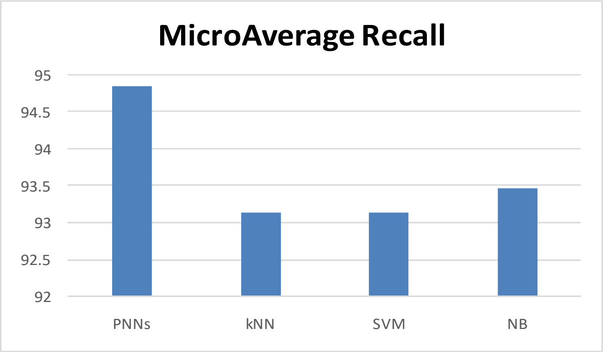

Figure 1. Spam Microaverage Recall

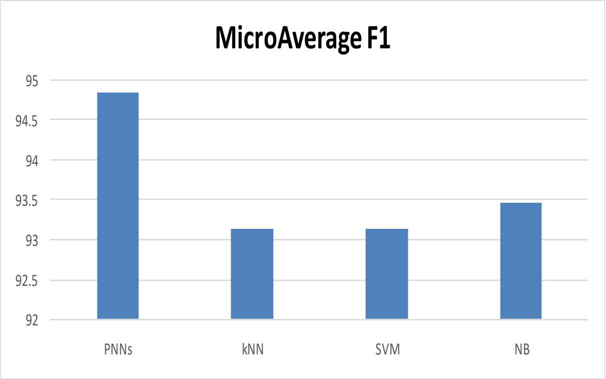

Figure 2. Spam Microaverage F1

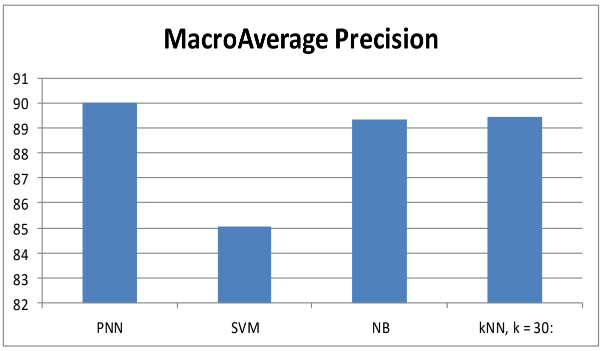

Figure 3. Spam Macroaverage Precision

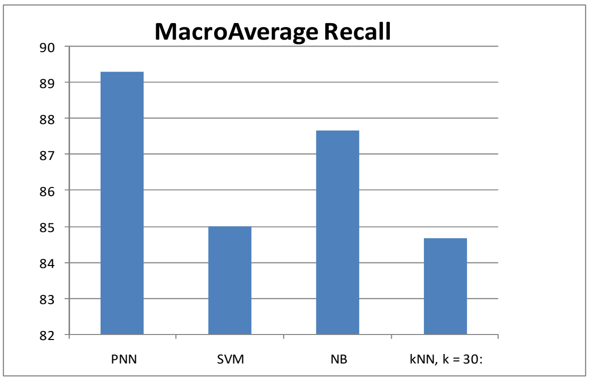

Figure 4. Spam Macroaverage Recall

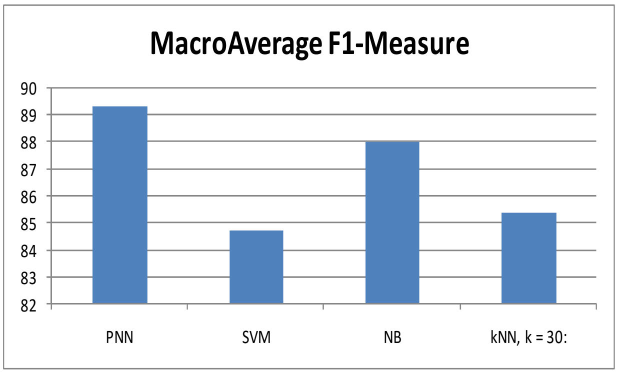

Figure 5. Spam Macroaverage F1

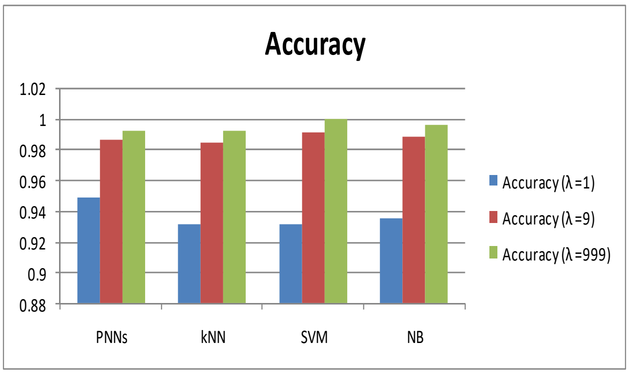

Figure 6. Spam Accuracy

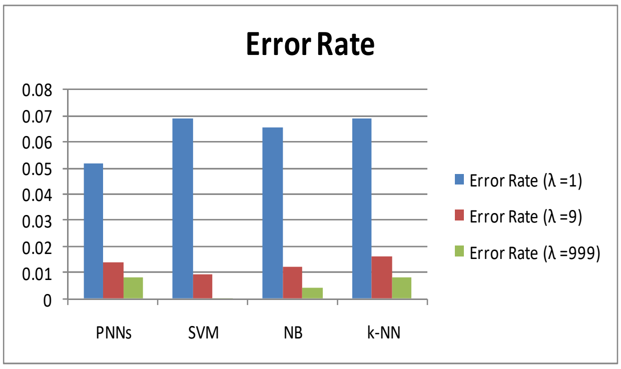

Figure 7. Spam Error Rate

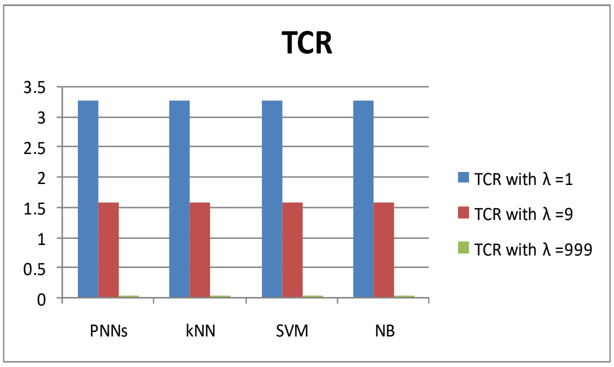

Figure 8. Spam TCR.

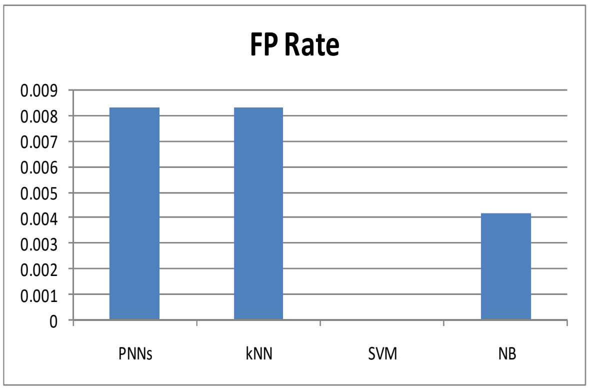

Figure 9. Spam FP rate

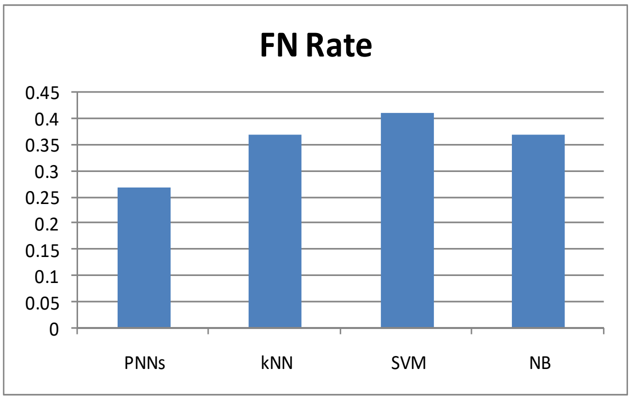

Figure 10. Spam FN rate

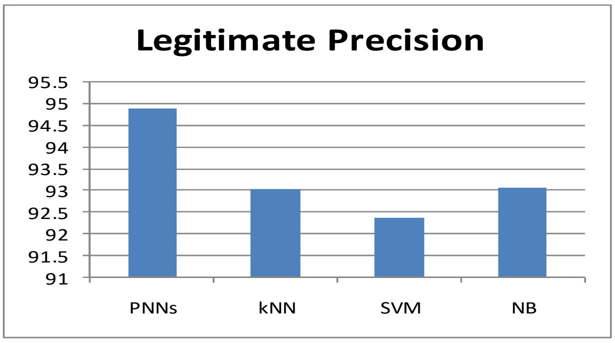

Figure 11. Legitimate Precision

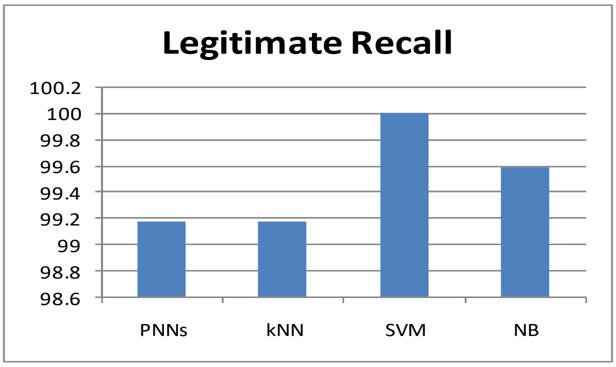

Figure 12. Legitimate Recall

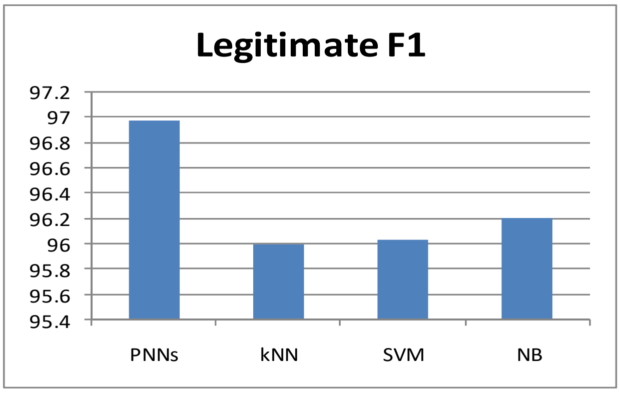

Figure 13. Legitimate F1

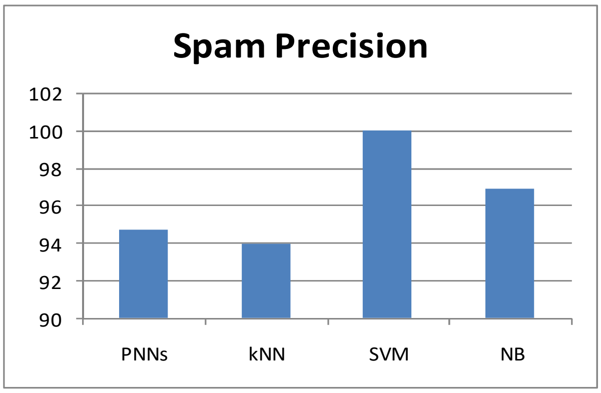

Figure 14. Spam Precision

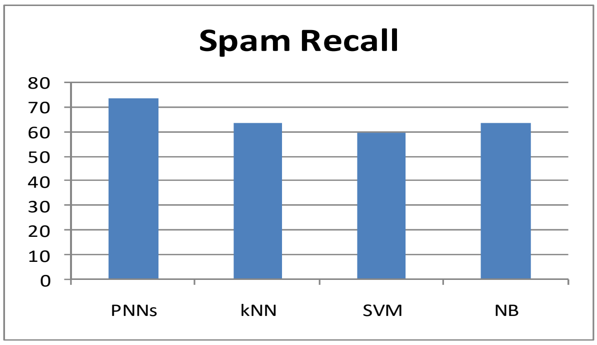

Figure 15. Spam Recall

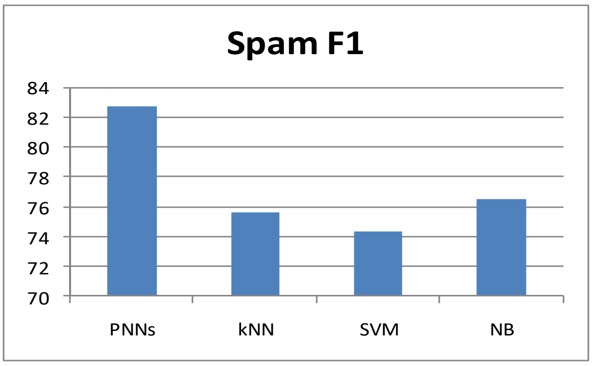

Figure 16. Spam F1

5. Conclusion

PNNs have proved in this research that it is a competitive spam e-mail filter using a very small but carefully selected subset of the corpus features. In the experiments conducted in this research, *PNNs* clearly outperformed the famous *SVM*, *k-NN* and *NB* spam filters in most of the performance measures, when the same settings were applied on all filters, including the pre-processing steps conducted on the email messages and the feature selection and reduction criteria. This indicates that *PNNs* is a promising spam filter which can be further investigated for commercial use, as it was able to record a distinguishable performance using only a very low number of features in classification. Our next step will be to directly compare *PNNs* with other famous spam filters on other available public email corpora.

References

[1]. Ling-Spam data set. (2015). Retrieved from: http://csmining.org/index.php/ling-spam-datasets.html [accessed 18 September 2019].

[2]. Porter Stemmer. (2015). Retrieved from: http://tartarus.org/martin/PorterStemmer/c.txt [accessed: 20 September 2019].

[3]. Fukuaga, K. (1990). Introduction to statistical pattern classification. *Patt. Recognit, 30*(7), 1145-1149.

[4]. Campbell, W. M., Assaleh, K. T., & Broun, C. C. (2001). A novel algorithm for training polynomial networks. *In International NAISO Symposium on Information Science Innovations ISI.*

[5]. Liu, C. L. (2006, August). Polynomial network classifier with discriminative feature extraction. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*(pp. 732-740). Springer, Berlin, Heidelberg.

[6]. Assaleh, K., & Al-Rousan, M. (2005). A new method for Arabic sign language recognition. *EURASIP Journal on Applied Signal Processing*, 2136-2145.

[7]. Al-Tahrawi, M. M., & Abu Zitar, R. (2008). Polynomial networks versus other techniques in text categorization. *International Journal of Pattern Recognition and Artificial Intelligence*, *22*(02), 295-322.

[8]. Al-Tahrawi, M. M., & Al-Khatib, S. N. (2015). Arabic text classification using Polynomial Networks. *Journal of King Saud University-Computer and Information Sciences*, *27*(4), 437-449.

[9]. Vapnik, V. (2013).*The nature of statistical learning theory*. Springer science & business media.

[10]. Vapnik, V. N. (1999). An overview of statistical learning theory. *IEEE transactions on neural networks*, *10*(5), 988-999.

[11]. Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In *European conference on machine learning* (pp. 137-142). Springer, Berlin, Heidelberg.

[12]. Kolcz, A., & Alspector, J. (2001). Svm-based filtering of email spam with content-specific misclassification costs, TextDM'2001 (IEEE ICDM-2001 Workshop on Text Mining).

[13]. Shen, Y., Sun, G., Qi, H., & He, X. (2010, December). Using feature selection to speed up online svm based spam filtering. In *2010 International Conference on Asian Language Processing* (pp. 142-145). IEEE.

[14]. Zheng, Z., Wu, X., & Srihari, R. (2004). Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, *6*(1), 80-89.

[15]. Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine learning*, *29*(2-3), 103-130.

[16]. Lewis, D. D., & Ringuette, M. (1994, April). A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval* (Vol. 33, pp. 81-93).

[17]. Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Sakkis, G., Spyropoulos, C. D., & Stamatopoulos, P. (2000). Learning to filter spam e-mail: A comparison of a naive bayesian and a memory-based approach. *arXiv preprint cs/0009009*.

[18]. Androutsopoulos, I., Koutsias, J., Chandrinos, K. V., & Spyropoulos, C. D. (2000, July). An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 160-167).

[19]. Metsis, V., Androutsopoulos, I., & Paliouras, G. (2006, July). Spam filtering with naive bayes-which naive bayes?. In *CEAS* (Vol. 17, pp. 28-69).

[20]. Kim, S. B., Seo, H. C., & Rim, H. C. (2003, July). Poisson naive Bayes for text classification with feature weighting. In *Proceedings of the sixth international workshop on Information retrieval with Asian languages-Volume 11* (pp. 33-40). Association for Computational Linguistics.