ISSN: 0067-2904

Spam Filtering Approach based on Weighted Version of Possibilistic c-Means

Sarab M. Hameed[*1], Marwan B. Mohammed[2]
[1]Department of Computer Science, College of Science, University of Baghdad, Baghdad, Iraq.
[2]Presidency of Al-Nahrain University, Al-Nahrain University, Baghdad, Iraq

Abstract

A principal problem of any internet user is the increasing number of spam, which became a great problem today. Therefore, spam filtering has become a research focus that attracts the attention of several security researchers and practitioners. Spam filtering can be viewed as a two-class classification problem. To this end, this paper proposes a spam filtering approach based on Possibilistic c-Means (PCM) algorithm and weighted distance coined as (WFCM) that can efficiently distinguish between spam and legitimate email messages. The objective of the formulated fuzzy problem is to construct two fuzzy clusters: spam and email clusters. The weight assignment is set by information gain algorithm. Experimental results on spam based benchmark dataset reveal that proper setting of feature-weight can improve the performance of the proposed spam filtering approach. Furthermore, the proposed spam filtering approach performance is better than PCM and Naïve Bayes filtering technique.

Keywords: Possibilistic c-Means (PCM) algorithm, Weighted PCM, Naïve Bayes, Spam filtering

اسلوب لتصفية البريد المزعج اعتمادا على نسخة موزونه من Possibilistic c-Means

سراب مجيد حميد[*1]، مروان بدران محمد[2]
[1]قسم علوم الحاسبات، كلية العلوم، جامعة بغداد، بغداد، العراق.
[2]رئاسة جامعة النهرين، جامعة النهرين، بغداد، العراق.

الخلاصة

المشكلة الرئيسية لمستخدمي الانترنت هو العدد المتزايد من البريد المزعج، والتي أصبحت مشكلة كبيرة اليوم. لذلك، أصبحت البحوث تركزعلى تصفية الرسائل غير المرغوب فيها والتي جذبت انتباه العديد من باحثين الامنية. يمكن اعتبار عملية تصفية البريد المزعج كمشكلة لتصنيف مجموعتين هذا البحثيقترح نهج لتصفية البريد المزعج على أساس Possibilistic c-Means والمسافة الموزنه التي يمكن أن تميز بكفاءة بين البريد المزعج والبريد الإلكتروني الشرعي. إن الهدف من استخدام خوارزمية التجمع الضبابي في الكشف هو تكوين مجموعتين من التجمع الضبابي هما: مجموعة البريد المزعج و مجموعة الرسائل الالكترونية. تحديد الوزن يتم عن طريق خوارزمية كسب المعلومات. النتائج التجريبية على مجموعة بيانات البريد المزعج القياسية اظهرت أن تحديد الوزن المناسب الى كل ميزة يمكن أن يحسن من أداء نهج

*Email: sarab_majeed@yahoo.com

تصفية البريد المزعج المقترحة. وعلاوة على ذلك، فإن أداء نهج تصفية المزعج المقترح هو أفضل من تقنية تصفية المزعج PCM وتقنية بايز البسيطة.

1. Introduction

In the last few decades, the electronic mail (email) became one of the most important ways of communication. Therefore, several people and companies attempt to send a vast amount of unsolicited messages to the massive number of users. This type of messages are called spam mail [1]. Spam is flooding the Internet with massive versions of a single message, in an attempt to oblige the message on people who could not refuse it [2]. Undoubtedly the reason to send those messages by email is easy communication methods, cost effectiveness [1, 2] and an import carrier for non-performing commercial advertising, hacker programs, the spread of the virus, and so on [3].

The spam mail has caused some problems. The first one, it causes loss of network resources, which is significant for network users. Moreover, practically it greatly affects the daily work for a lot of users; the people are wasting a lot of time dealing with spam, there are many spam mails which attract users, but it may in fact contain unexpected malicious attachments which would seriously crack the user's system [4].

There are various techniques to anti-spam [2], but usually their techniques vary daily, whatever anti-spam technology used; it must be capable to adapt rapidly. There are three important characteristics to reach a good anti-spam technique: firstly, it will accurately classify spam and legitimate mail; secondly, it will be well adaptable, and finally, it will be easily scalable [5] .

Usually, spam has unqualified or no absolute definition to distinguish it from legitimate emails. Hence, the discipline of Machine Learning (ML) has recently engaged considerable attention in the design of effective spam filtering functions.

In 2011 [3], two methods were proposed. The first method is used to calculate the similarity between semantic bodies based on sentence similarity and the second one is fuzzy clustering method based on the semantic. The reason of using fuzzy clustering was to solve the problem imprecision and fuzziness exist in spam. The results show that the method based on a semantic body of spam filtering is feasible and has good application prospects.

In 2013 [6], a spam word ranking and fuzzy rules are used to put emails in groups regarding the threats of each word. The proposed work used two sets of linguistic terms for ranking and classifying spam mails. This method has extracted only the features from the content of an email instead of extracting all the features from the mail. The results of ranking and fuzzy rules are outperform ranking and classifying of spam words.

In 2013 [7], FCM was applied in verifying spams. In fuzzy clustering each feature could join to its similar cluster regardless with different membership degrees (between 0 and 1). It is suitable for both small and large datasets. They used two dataset: Lemm-Stop dataset and own dataset to classify the spam.

In 2013 [8] , Optical Back Propagation (OBP) technique was proposed to identify whether a message is spam or email based on the content of the message. The performance of the proposed OBP-based spam is reasonable for different sizes of training and testing dataset.

In 2014 [9] , a hybrid spam filtering mechanism based on K-means clustering and SVM was proposed. The evaluation of the hybrid mechanism carried out using a spam based standard dataset. The hybridization mechanism results in decreasing training time and increasing accuracy of SVM classifier.

In 2015 [10], S. M. Hameed, M.B. Mohammed, and B. A. Attea present spam filtering methodology based on the concept of fuzziness mean, particularly, fuzzy c-means (FCM) algorithm and information gain algorithm. Experimental results on spam based dataset point out the proposed spam filtering is more efficient than with the known Naïve Bayes filtering technique

In this paper, one of the fuzzy clustering family named possiblictic c-means algorithm (PCM) will be modified to design spam filtering that can efficiently distinguish between spam and legitimate email messages. The remainder of this paper is organized as follows. Section 2 describes the basic concepts behind achieving PCM. Section 3 gives a brief description on the spam based dataset. Section 4 illustrates the suggested spam filtering algorithm based on the proposed weighted PCM.

Section 5 illustrates experimental results. Finally, section 6 presents conclusions carried out after this work.

2. Possibilistic c-Means Algorithm

This section presents a brief background on PCM. PCM is unsupervised clustering algorithm For each data point or sample, s_i, in a given data set $S = \{s_1, s_2, \dots, s_n\}$, PCM computes its possibility degrees to each of k clusters. Each of the clusters, $c_i, 1 \le i \le k$, is represented by its center (or prototype), v_i. Thus, a complete set of k prototypes $V = \{v_1, v_2, \dots, v_k\}$ is to be produced by PCM. At the beginning of PCM, the values of these prototypes are selected randomly. Then, according to the Euclidean distance (d^2), each sample vector $s_j, 1 \le j \le n$ is assigned a possibility degrees, $t_{ij} \in [0,1]$, to each cluster v_i. Thus, PCM can construct a $k \times n$ matrix $U = [t_{ij}]$. For fuzzy clustering, possibility degrees should be summed up to 1 as in Eq. (1) [11].

$\forall_i,\ 1 \le i \le k$:

$$\sum_{j=1}^{n} t_{ij} = 1 \quad (1)$$

PCM algorithm aims to minimize the function formulated in Eq. (2) [11]:

$$Min\, J_m(S,U,V) = \sum_{i=1}^{k}\sum_{j=1}^{n} t_{ij}^{\delta}\, d^2(s_j - v_i) + \sum_{i=1}^{k} \eta_i \sum_{j=1}^{n}(1 - t_{ij})^{\delta} \quad (2)$$

Where

The scale parameter η_i can be obtained from the average possibilistic intra-cluster distance of cluster i as in Eq. (3) [11]

$$\eta_i = \frac{\sum_{j=1}^{n} t_{ij}^{\delta} d^2(s_j, v_i)}{\sum_{j=1}^{n} t_{ij}^{\delta}} \quad (3)$$

and

$\delta \in [1, \infty)$ is a weighting factor called the possibilistic parameter.

Since, the objective function $\text{Min}\, J_m(S,U,V)$ cannot be minimized directly, an iterative algorithm is used to iteratively optimize possibility degrees and cluster centers by updating $t_{ij}$and v_i using Eq. (4) and Eq. (5) respectively [11].

$$v_i = \frac{\sum_{j=1}^{n}(t_{ij})^{\delta} s_j}{\sum_{j=1}^{n}(t_{ij})^{\delta}} \quad (4)$$

$$t_{ij} = \frac{1}{1+\left(\frac{d^2(s_j,v_i)}{\eta_i}\right)^{\frac{1}{\delta-1}}} \quad (5)$$

In the objective function (2), the first term demands that the distances from data points to the prototypes be as low as possible, whereas the second term forces the $t_{ik}$to be as large as possible, thus avoiding the trivial solution.

3. Dataset Description

Spambase dataset is used as the input space $S = \{s_1, s_2, \dots, s_n\}$ of messages. Corpora dataset is a widely used spambased dataset created in 1999, by M. Hopkins, E. Reeber, G. Foreman and J. Suermondt of Hewlett Packards Labs. This dataset consists of a table of 4601 rows (or records), each of 58 columns. Each row corresponds to one random message, while each column represent one attribute or feature characterizing the message at the corresponding row. The first 57 features are variables and the last one indicates if it spam (1) or legitimate email (0). The total number of spam, n_s, in this dataset is 1813 (forming 39.4% of the total dataset), while the total number of legitimate emails, n_e, is 2788 (i.e., forming 60.6% of the total dataset) [12].

Thus, corpora dataset can be formally described as $S = \{s_1, s_2, \dots, s_n\}$, where $n = n_s + n_e = 4601$. Moreover, each message, $s_i \in S$, can be formulized as:

$\forall i \in \{1, \dots, n\}$

$s_i = \{s_{i1}, s_{i2}, \dots, s_{i57}\}$

4. Weighted PCM for Spam Filtering Model

In this section, a weighted version of PCM as a spam filtering model is proposed. The basic idea behind the proposed Weighted PCM (WPCM) is to replace Euclidean distance (d^2), used in PCM algorithm to measure the similarity between samples by a weighted Euclidean distance (d_w^2).

In the computation of the traditional Euclidean distance, all features have an equal weight. On the other hand, in the proposed weighted Euclidean distance, different features have different weights, in the range [0,1] specifying different importance of the features with respect to the distance

computation. Each feature is assigned with its weight obtained from the information gain algorithm. Formally speaking, the formulation of the weighted Euclidean distance (d_w^2) can be described as in Eq. (6).

$$d_w^2(s_j, v_i) = \sqrt{\sum_{k=1}^{|\mathcal{F}|}(w_k)^2 * (s_{jk} - v_{ik})^2} \quad (6)$$

Where

$W = \{w_1, w_2, \dots, w_{|\mathcal{F}|}\}$, is the weight vector.

Each element of W represents the importance degree related to each feature. Larger w_k is more significant the k^{th} feature is in WPCM. On the other hand, the lower value of w_k is less significant the k^{th} feature is in WPCM. The objective function of the WPCM can be formulated in Eq. (7):

$$\text{Min } J_m^w(S, T, V) = \sum_{i=1}^{k}\sum_{j=1}^{n} t'^{\delta}_{ij}\, d_w^2(s_j - v_i) + \sum_{i=1}^{k}\eta_i \sum_{j=1}^{n}(1 - t'_{ij})^{\delta} \quad (7)$$

Where the suggested scale parameter η_i' at the i^{th}cluster and suggested possibility degrees $t'_{ij}$of sample s_j in cluster v_i are calculated as in equation (8) and (9) respectively.

$$\eta'_i = \frac{\sum_{j=1}^{n_t} t^{\delta}_{ij} d_w^2(s_j, v_i)}{\sum_{j=1}^{n_t} t^{\delta}_{ij}} \quad (8)$$

$$t'_{ij} = \frac{1}{1 + \left(\frac{d_w^2(s_j, v_i)}{\eta_i}\right)^{\frac{1}{\delta - 1}}} \quad (9)$$

The suggested WPCM is utilized for spam filtering coined as WPSF, which consists of two modules: training module ($WPSF_{trn}$) and testing module ($WPSF_{tst}$).

First, the dataset S is preprocessed to remove irrelevant or weak features out of the total 57 features. Removing irrelevant features, or in other words, selecting a distinguished feature set, $\mathcal{F}$, out of the complete 57 features, is carried out by adopting information gain algorithm. The information gain value $W = \{w_1, w_2, \dots, w_{|\mathcal{F}|}\}$ for each feature obtained from information gain algorithm are fed to testing module. Formally speaking, if $\mathbb{F} = \{\mathbb{F}_1, \mathbb{F}_2, \dots, \mathbb{F}_{57}\}$ is the complete feature set, then, $\mathcal{F} \subseteq \mathbb{F}$.

Since features in the feature set $\mathcal{F}$ can normally have different scales of values, then, the second preprocessing step is to normalize the values of these features to be in the range of $[0,1]$.

Now, the input set of samples, S', is ready for handling by training of WPSF. Let the size of the trained dataset is n_t, i.e., $S' = \{s'_1, s'_2, \dots, s'_{n=n_t}\}$. The purpose of the training module is to construct two clusters, namely, spam cluster, $c1$, and legitimate email cluster, $c2$. The formation of these two clusters can be achieved by specifying the prototype value (i.e., center) of each one. After preprocessing, the distinguished feature set $\mathcal{F}$ can be used by WPCM to define the prototype of each cluster. The role of the training module is to train, according to a set $S = \{s_1, s_2, \dots, s_n\}$ of a priori classified messages, two prototype vectors $V = \{v_1, v_2\}$. The generated prototype vectors should be correct enough to meet the appropriate spam and legitimate email cluster centers c_1 and c_2, respectively. The main steps of training module of WPSF are presented in algorithm 1.

On the other hand, the goal of the testing module of WPSF as summarized in algorithm (2) is to make, based on the trained prototype vectors V produced from the training module, a binary classification decision on the incoming message(s).The testing module of WPSF will assign label C_{spam} or C_{email} to the tested message s_j.

5. Experimental Results

This section experimentally tests the effectiveness of the proposed WPSF. A set of experiments and comparison have been conducted to show the applicability of WPSF on clustering spam and legitimate email messages.

5.1 Evaluation Metrics

The performance of WPSF is evaluated using the following three criteria.

1. Accuracy (Acc): this measure reflects the percentage of predictions that are correct [1]. The formula for calculating this measure is given as in Eq. 10.

$$\text{Acc} = \frac{TP+TN}{TP+TN+FP+FN} \quad (10)$$

Where

TP True Positive is considered for spam that is correctly classified.

FP False Positive occurs when legitimate email is misclassified as spam.

TN True Negative considers legitimate message that is correctly classified and
FN False Negative occurs when the spam is misclassified as legitimate email.

2. Spam Recall (SR**):** is defined as the ratio of the number of correctly detected spam [1] . The formula for calculating this measure is:

$$\mathrm{SR} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (11)$$

3. Spam Precision (SP**):** is the percentage of the predicted positive cases that are correct [1] . The formula for calculating this measure is:

$$SP = \frac{TP}{TP+FP} \quad (12)$$

Algorithm 1 *: $WPCM$ for $WPSF_{trn}$*

Input:

- *Number of samples in the training set* n_t.
- *Dataset*: $S' = \{s'_1, s'_2, \dots, s'_{n=n_t}\}$
- *Number of selected features* $|\mathcal{F}| = \{13,17,23,29,57\}$
- *Number of clustersc* $= 2$.
- *Possiblistic parameter (δ)*
- *Set iteration number* $r = 0$
- *Stopping criterion* $\varepsilon = 0.01$
- *Weighted for each feature* $W = \{w_1, w_2, \dots, w_{|\mathcal{F}|}\}$

Output:

- Prototype vector for spam cluster, $v_1 = \{v_{11}, v_{12}, \dots, v_{1|\mathcal{F}|}\}$
- Prototype vector for legitimate email cluster, $v_2 = \{v_{21}, v_{22}, \dots, v_{2|\mathcal{F}|}\}$

Steps:

1. Initialize prototype vectors v_1^0, v_2^0 to value as produced by the PSF_{trn}

2. Assign typicality degrees t'^0_{ij} to value of t_{ij} calculated from PSF_{trn}, $\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\}$

3. Calculate Euclidean distance between each sample, s'_j, and the prototype of the two clusters v_1^r, and v_2^r

$\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\} \wedge \forall k \in \{1, \dots, |\mathcal{F}|\}$

$$d_w^2(s_j, v_i) = \sqrt{\sum_{k=1}^{|\mathcal{F}|} (w_k)^2 * (s_{jk} - v_{ik})^2}$$

4. Compute scale parameter η'_i

$\forall i \in \{1,2\}$

$$\eta'_i = \frac{\sum_{j=1}^{n_t} t^{\delta}_{ij} d_w^2(s_j, v_i)}{\sum_{j=1}^{n_t} t^{\delta}_{ij}}$$

5. Update typicality degrees t'_{ij} of each sample s_j in cluster $v_i$as:

$\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\}$

$$t'^{r+1}_{ij} = \frac{1}{1 + \left(\frac{d^2\left(s'_j, v_i\right)}{\eta'_i} \right)^{\frac{1}{\delta - 1}}}$$

6. Update prototype values v_1 and v_2 of the two clusters

$\forall i \in \{1,2\}$

$$v_i^{r+1} = \frac{\sum_{j=1}^{n_t} \left(t'_{ij}\right)^{\delta} S'_j}{\sum_{j=1}^{n_t} \left(t'_{ij}\right)^{\delta}}$$

7. If $max\{|t'^{r+1}_{ij} - t'^{r}_{ij}|\} < \varepsilon$, $\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\}$ then stop, else increment iteration number, r, by one and go to step 3.

Algorithm 2: *: $WPCM$ for $WPSF_{tst}$*

Input:

- *Number of samples in testing dataset* n_{tst}
- *Dataset*: a set $S = \{s_1, \dots, s_{ntst}\}$
- *Number of selected features* $|\mathcal{F}| = \{13,17,23,29,57\}$
- *Number of clustersc* $= 2$.
- *Possiblistic parameter (δ)*
- *Weighted for each feature* $W = \{w_1, w_2, \dots, w_{|\mathcal{F}|}\}$

Output:

- *Classified Dataset*

Steps:

1. Initialize prototype vectors v_1, v_2 to value calculated in $WPSF_{trn}$

2. Calculate the Euclidean distance between each sample, s_j, and the prototype of the two clusters v_1, and v_2.

$\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_{tst}\} \wedge \forall k \in \{1, \dots, |\mathcal{F}|\}$

$$d_w^2(s_j, v_i) = \sqrt{\sum_{k=1}^{|\mathcal{F}|} (w_k)^2 * (s_{jk} - v_{ik})^2}$$

3. Compute scale parameter $\eta'_i, \forall i \in \{1,2\}$

$$\eta'_i = \frac{\sum_{j=1}^{n_{tst}} t^{\delta}_{ij} d_w^2(s_j, v_i)}{\sum_{j=1}^{n_{tst}} t^{\delta}_{ij}}$$

$For\ j = 1\ to\ n_{tst}$

Begin

4. Compute two typicality values t'_{1j} and t'_{2j}

$$t'_{1j} = \frac{1}{1 + \left(\frac{d_w^2(s_j, , v_1)}{\eta'_1}\right)^{\frac{1}{\delta - 1}}}$$

$$t'_{2j} = \frac{1}{1 + \left(\frac{d^2(s_j, v_2)}{\eta'_2}\right)^{\frac{1}{\delta - 1}}}$$

5. Assign label C_{Spam} or C_{email} for tested incoming email s_j $WPSF_{tst}(s_j, \{v_1, v_2\}) = \begin{cases} C_{email}\ if\ t'_{1j} > t'_{2j} \\ C_{spam}\ otherwise \end{cases}$

End

5.2 Training and Testing Datasets

In the training module, a training dataset is divided into seven groups; each contains distinct samples selected randomly from the spam-based dataset. On the other hand, the testing dataset is divided into four groups, and each one contains distinct samples selected randomly from the remaining spam-based dataset. Tables- (1, 2) quantify the number of samples in the training and testing dataset groups.

The parameters WPSF are set as follows:

1. The initial value of the centroid of spam and legitimate clusters email are set to values produced from FCM.
2. The Initial value of the suggested typicality degree are set to values of membership degree calculated from FCM
3. Stopping criterion is $\varepsilon = 0.01$
4. The value of δ is set to 3.

Table 1- Training Dataset Groups

Group #	Number of Samples	Email Samples	Spam Samples
trn_1	300	180	120
trn_2	600	360	240
trn_3	900	540	360
trn_4	1200	720	480
trn_5	1500	900	600
trn_6	1800	1080	720
trn_7	2100	1260	840

Table 2- Testing Dataset Groups

Group #	Number of Samples	Email Samples	Spam Samples
tst_1	300	180	120
tst_2	600	360	240
tst_3	900	540	360
tst_4	1200	720	480

5.3 Impact of Number of selected features

The performance of WPSF is affected by increasing or decreasing number of selected features. Table- 3 presents accuracy result of WPSF using different percentage of information gain, i.e., different features. Five setting are experimented with. 100% (i.e., the complete set of 57 features are used), 50% of information gain (i.e., 29 features are selected), 40% (i.e., 23 features are selected), 30% of information gain (i.e. 17 features are selected) and 20% (i.e. 13 features are selected). The results in Table- 3 clarifies that using 50%, 40%, 30% or 20% WPSF can provide better compromise between WPSF's accuracy and computation cost.

The performance of WPSF is compared with PCM for spam filtering coined as PSF and Naïve Bayes (NB) for spam filtering in terms of accuracy, spam precision and spam recall. Figures-(1- 15) depict the accuracy, precision and spam recall of WPSF, PSF and NB with 100% , 50%, 40%, 30% and 20% of features when seven training datasets groups and four testing datasets groups are used**.** The results reveal that the WPSF are higher accuracy than PSF and NB regardless of number of samples in training and testing dataset groups. The spam recall of WPSF are higher than NB in all training and testing groups except in tst_4, NB's spam recall is better than or equal to spam recall of WPSF. These results reflect the ability of NB algorithm to bias towards maximizing spam recall at the expense of accuracy and precision. On the other hand, the proposed WPSF tends to maximize both accuracy and precision at the expense of spam recall function. To this end, one can say that WPSF model is more efficient than NB and PSF algorithm.

Table 3- Impact of features Number on WPSF's Accuracy.

Training Dataset Groups	Percentage	Acc%				Average
		tst_1	tst_2	tst_3	tst_4	
trn_1	**100%**	100.00	100.00	100.00	99.58	99.90
	50%	100.00	100.00	100.00	99.67	99.92
	40%	100.00	100.00	100.00	99.67	99.92
	30%	100.00	100.00	100.00	99.50	99.88
	20%	100.00	100.00	100.00	99.50	99.88
trn_2	**100%**	100.00	100.00	100.00	99.17	99.79
	50%	100.00	100.00	100.00	98.67	99.67
	40%	100.00	100.00	100.00	98.75	99.69
	30%	100.00	100.00	100.00	98.58	99.65
	20%	100.00	100.00	100.00	98.67	99.67

trn_3	**100%**	100.00	100.00	100.00	99.00	99.75
	50%	100.00	100.00	100.00	98.83	99.71
	40%	100.00	100.00	100.00	99.00	99.75
	30%	100.00	100.00	100.00	98.92	99.73
	20%	100.00	100.00	100.00	98.67	99.67
trn_4	**100%**	100.00	100.00	100.00	99.00	99.75
	50%	100.00	100.00	100.00	99.08	99.77
	40%	100.00	100.00	100.00	99.67	99.92
	30%	100.00	100.00	100.00	98.92	99.73
	20%	100.00	100.00	100.00	99.00	99.75
trn_5	**100%**	100.00	100.00	100.00	98.83	99.71
	50%	100.00	100.00	100.00	98.67	99.67
	40%	100.00	100.00	100.00	98.92	99.73
	30%	100.00	100.00	100.00	98.58	99.65
	20%	100.00	100.00	100.00	98.50	99.63
trn_6	**100%**	100.00	100.00	100.00	99.50	99.88
	50%	100.00	100.00	100.00	99.75	99.94
	40%	100.00	100.00	100.00	99.92	99.98
	30%	100.00	100.00	100.00	99.92	99.98
	20%	100.00	100.00	100.00	99.17	99.79
trn_7	**100%**	100.00	100.00	100.00	99.75	99.94
	50%	100.00	100.00	100.00	99.92	99.98
	40%	100.00	100.00	100.00	99.92	99.98
	30%	100.00	100.00	100.00	99.92	99.98
	20%	100.00	100.00	100.00	99.92	99.98

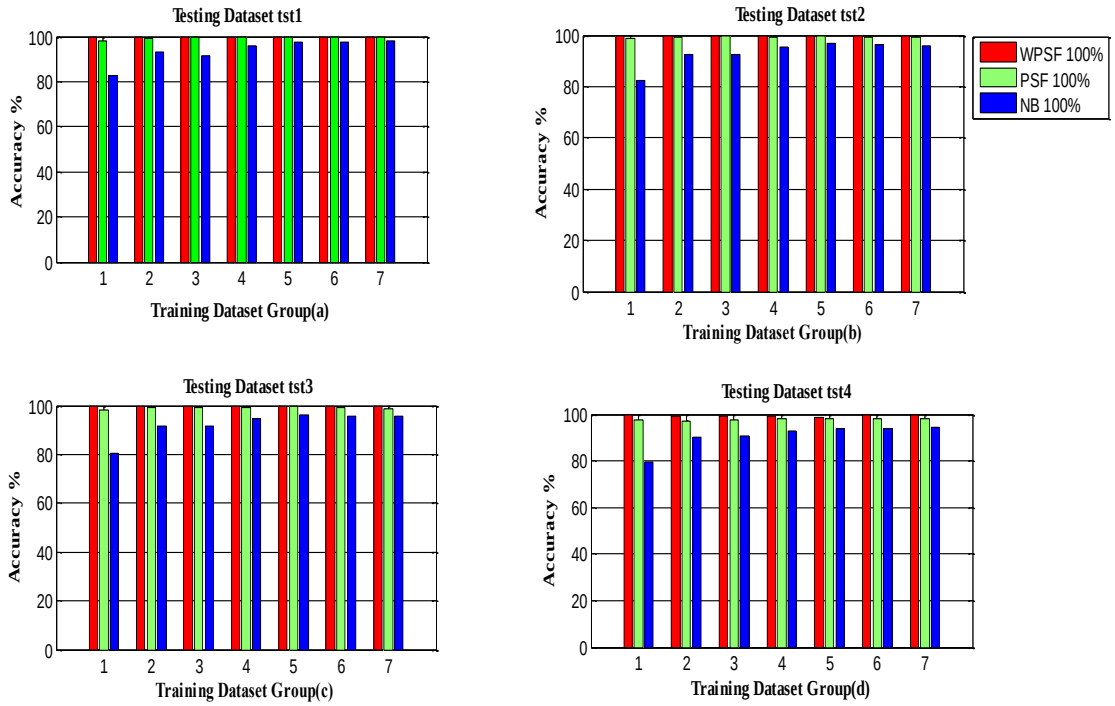

Figure 1- Accuracy for WPSF, PSF and NB with 57 features

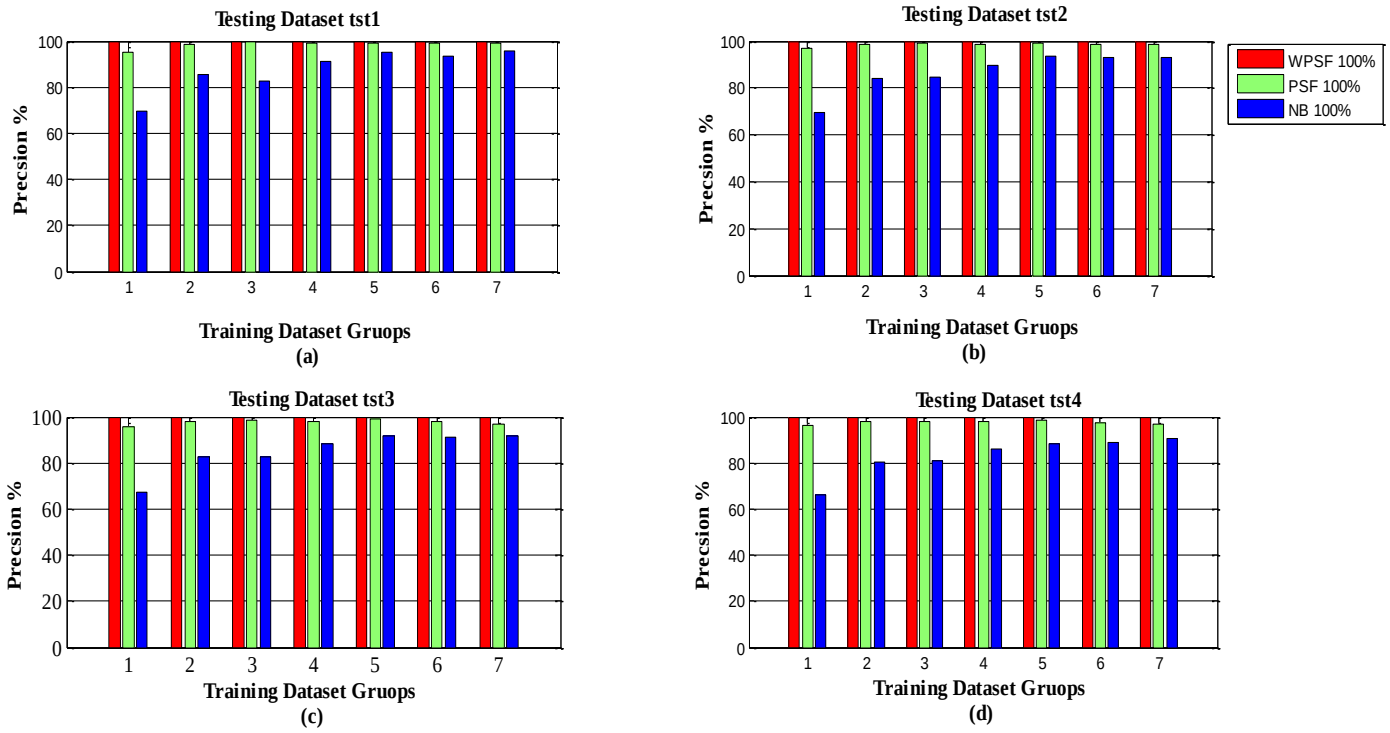

Figure 2- Precision for WPSF, PSF, and NB with 57 Features.

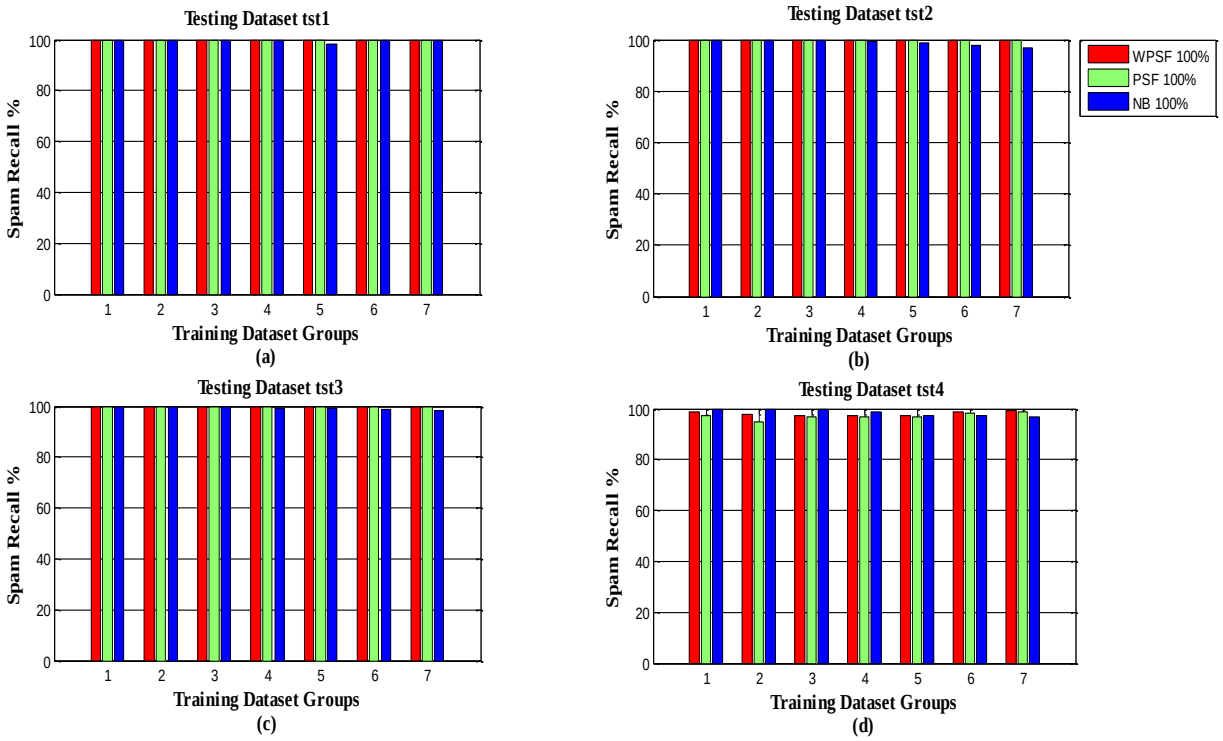

Figure 3- Spam Recall for WPSF, PSF and NB with 57 Features

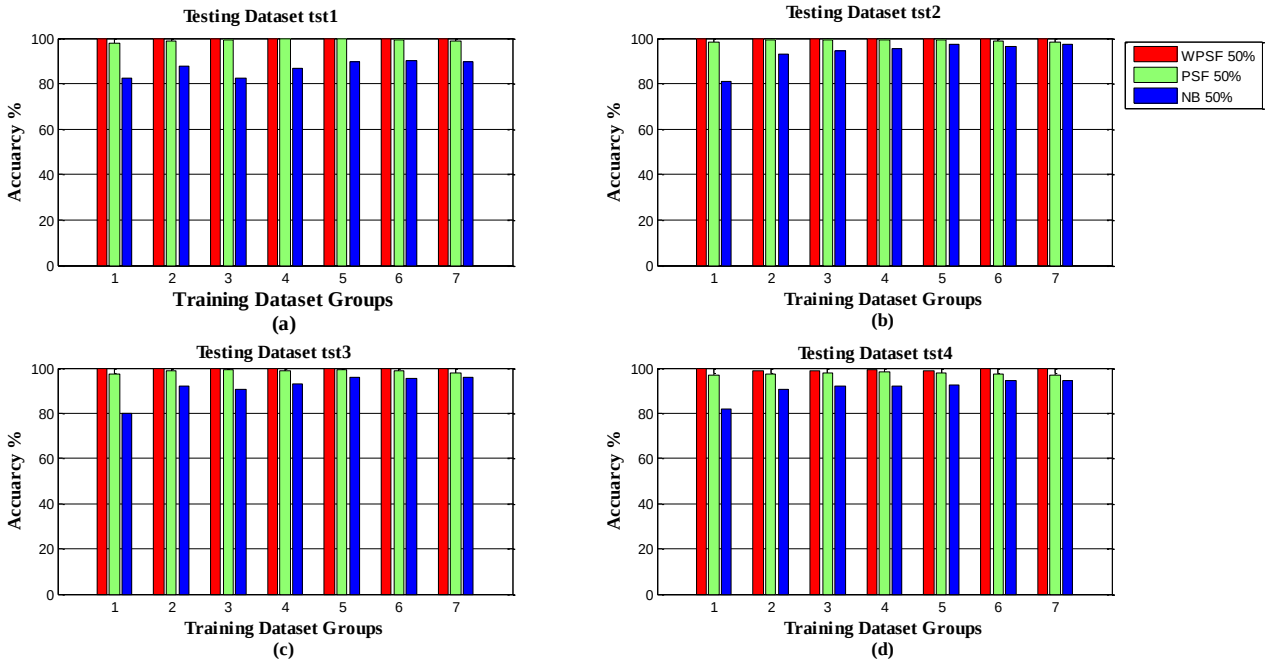

Figure 4- Accuracy for WPSF, PSF and NB with 29 Features.

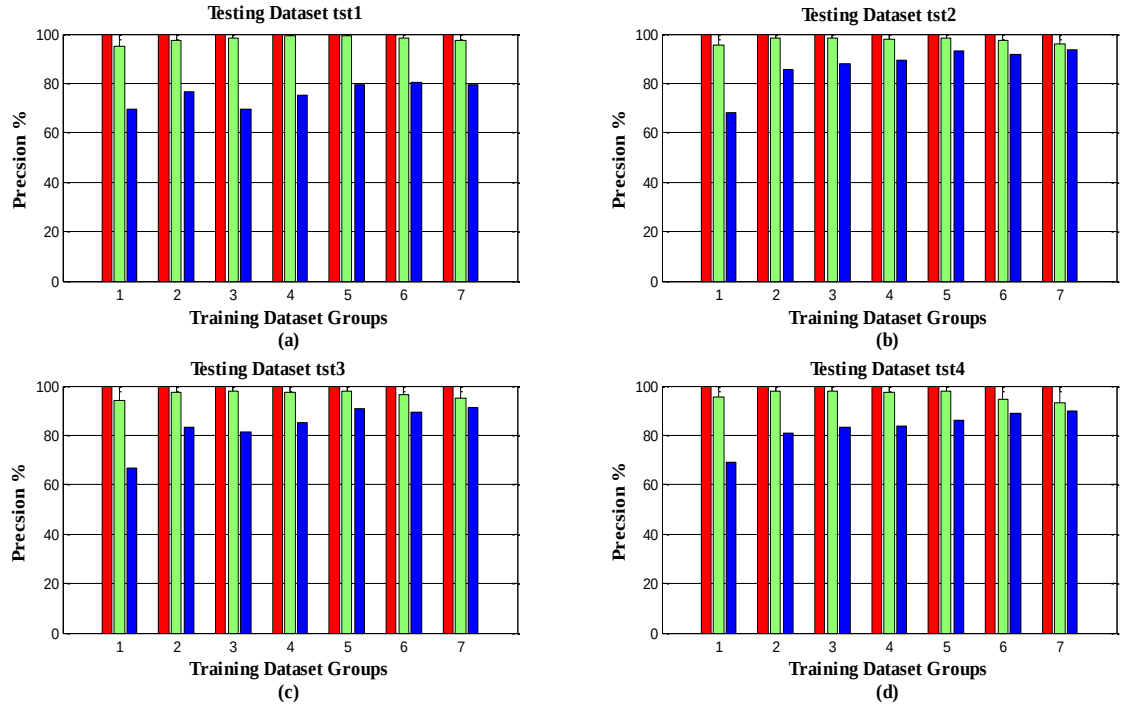

Figure 5- Precision WPSF, PSF and NB with 29 Features

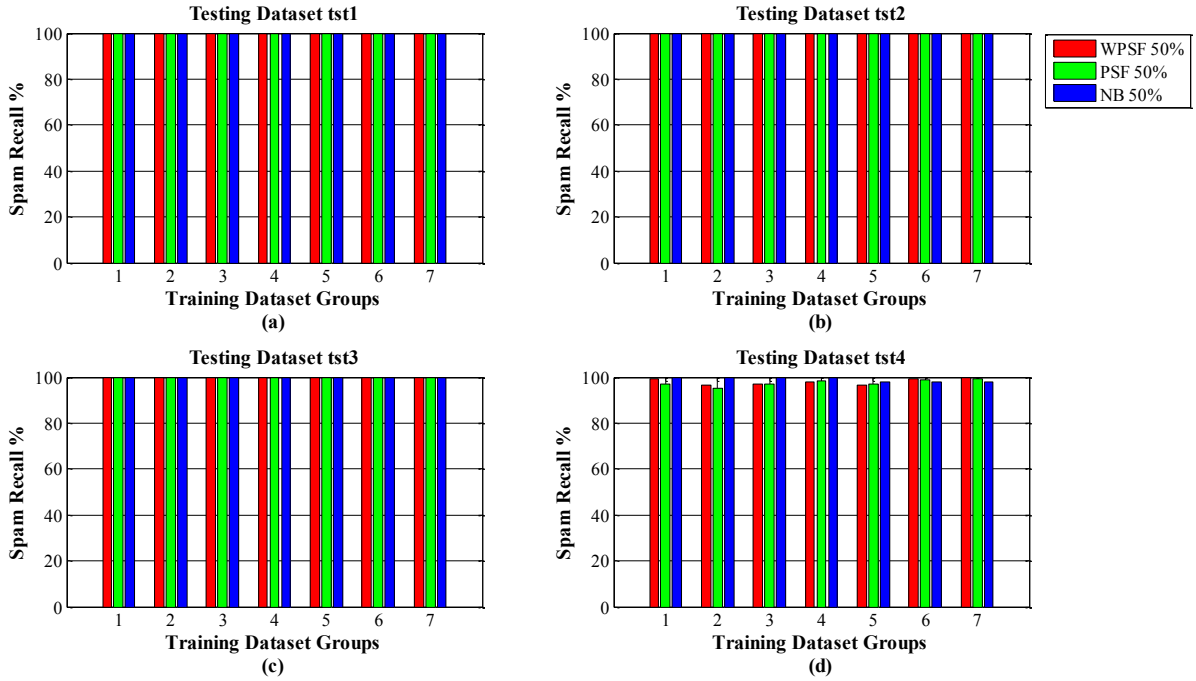

Figure 6-Spam Recall for WPSF, PSF and NB with 29 Features.

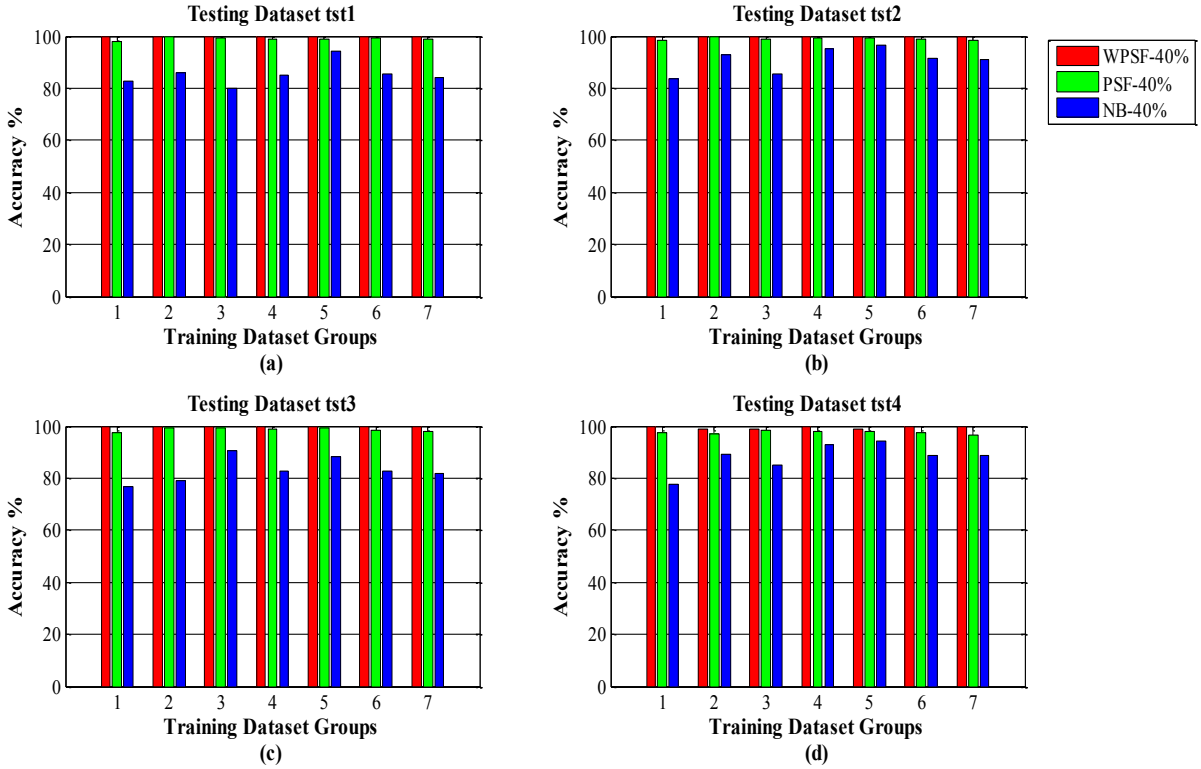

Figure 7- Accuracy for WPSF, PSF and NB with 23 Features.

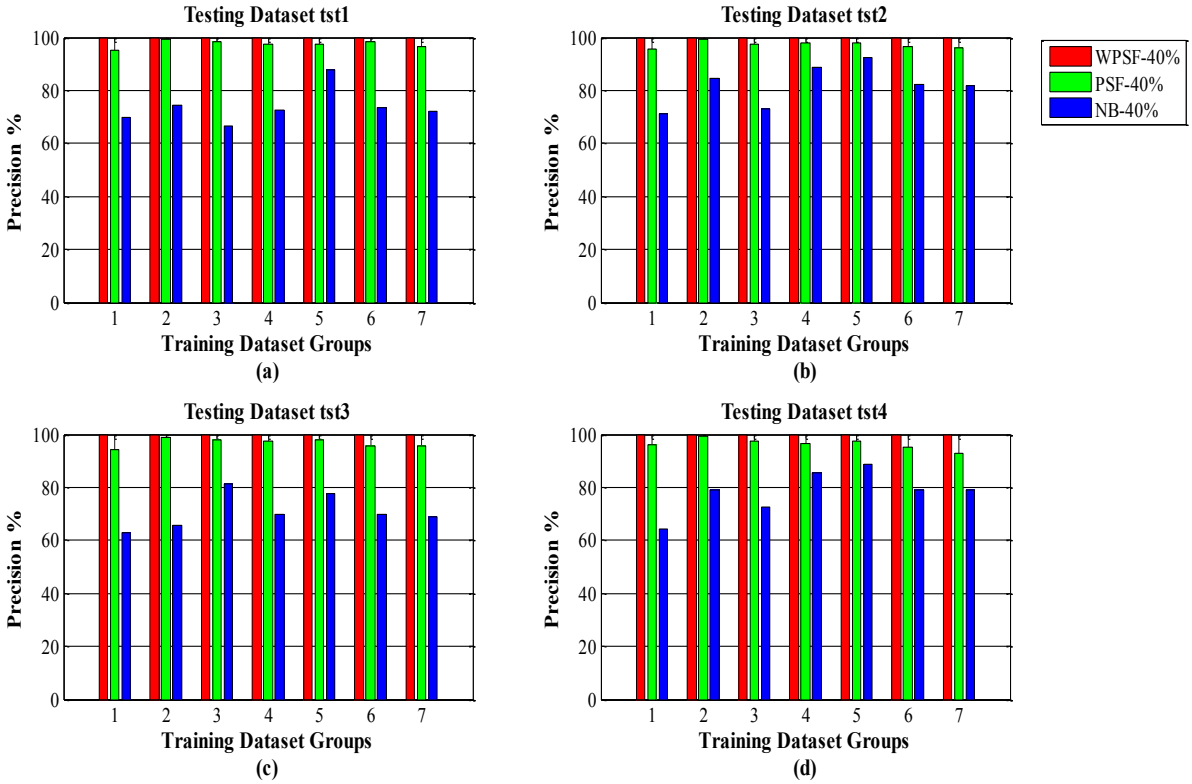

Figure 8- Precision for WPSF, PSF and NB with 23 Features

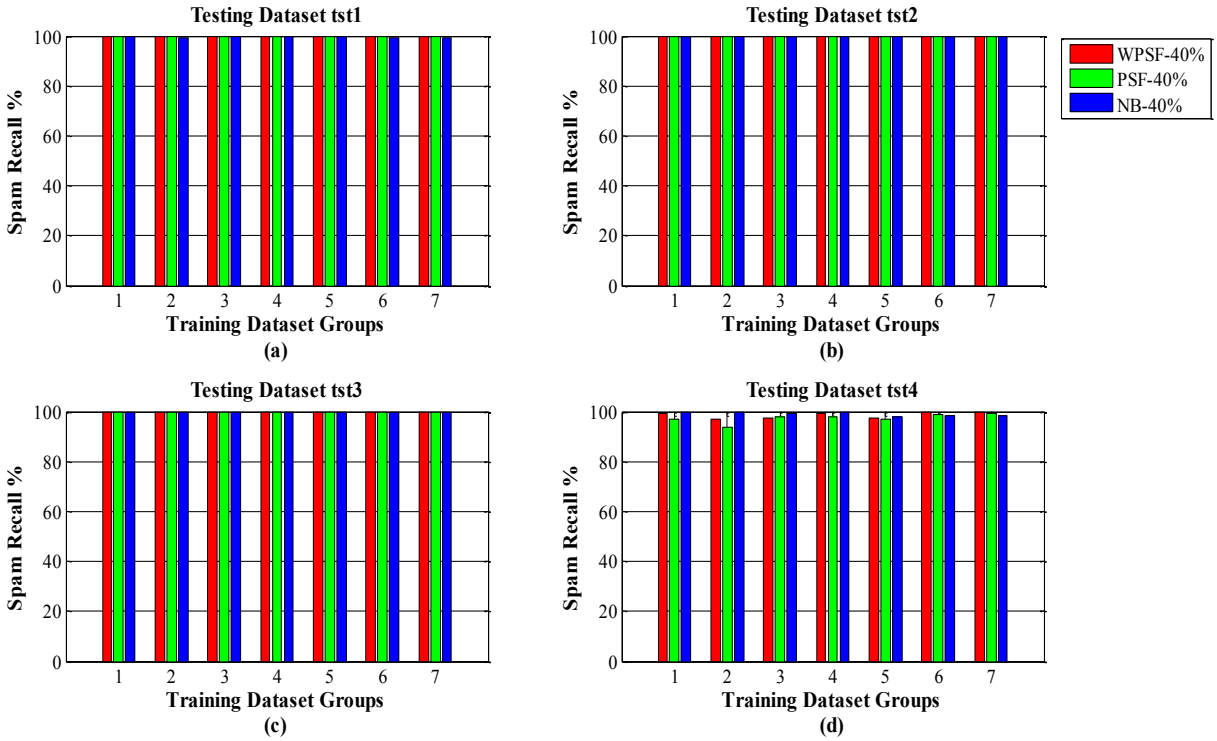

Figure 9- Spam Recall for WPSF, PSF and NB with 23 Features

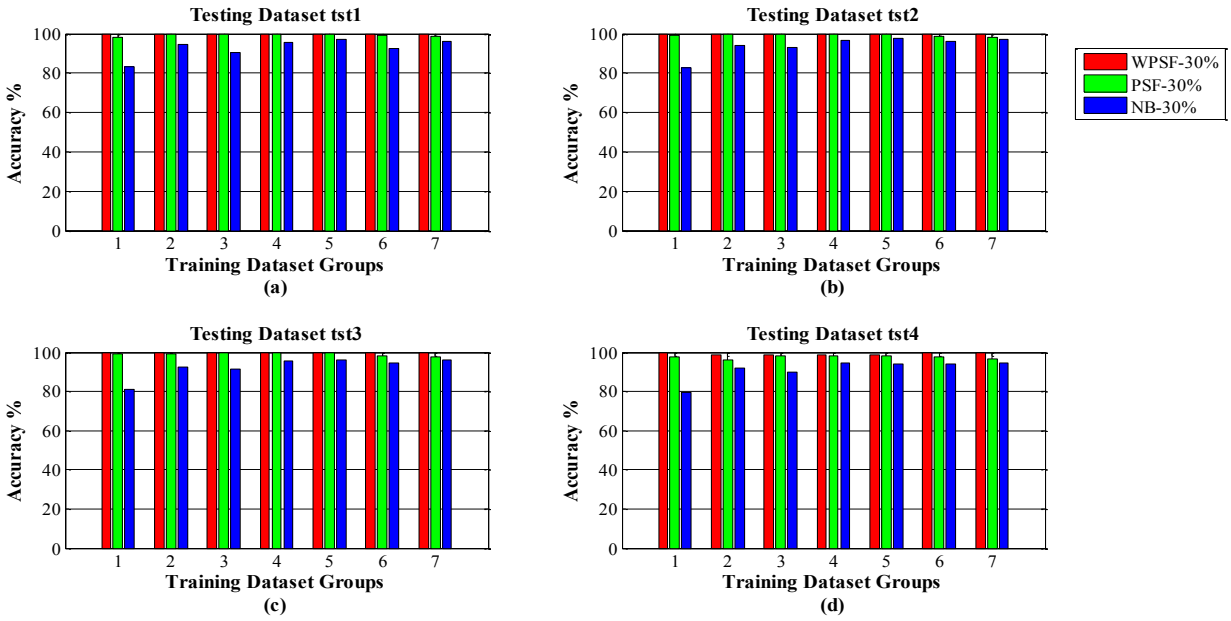

Figure 10- Accuracy for WPSF, PSF and NB with 17 Features.

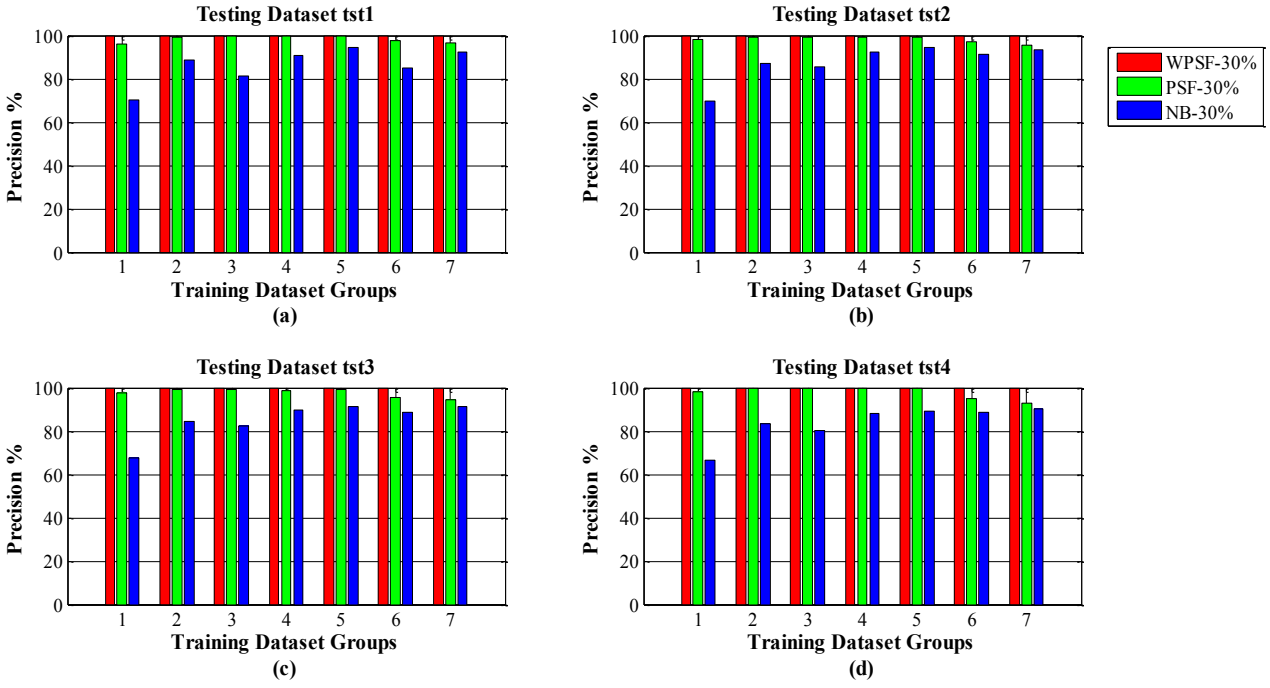

Figure 11-Precision for WPSF, PSF and NB with 17 Features

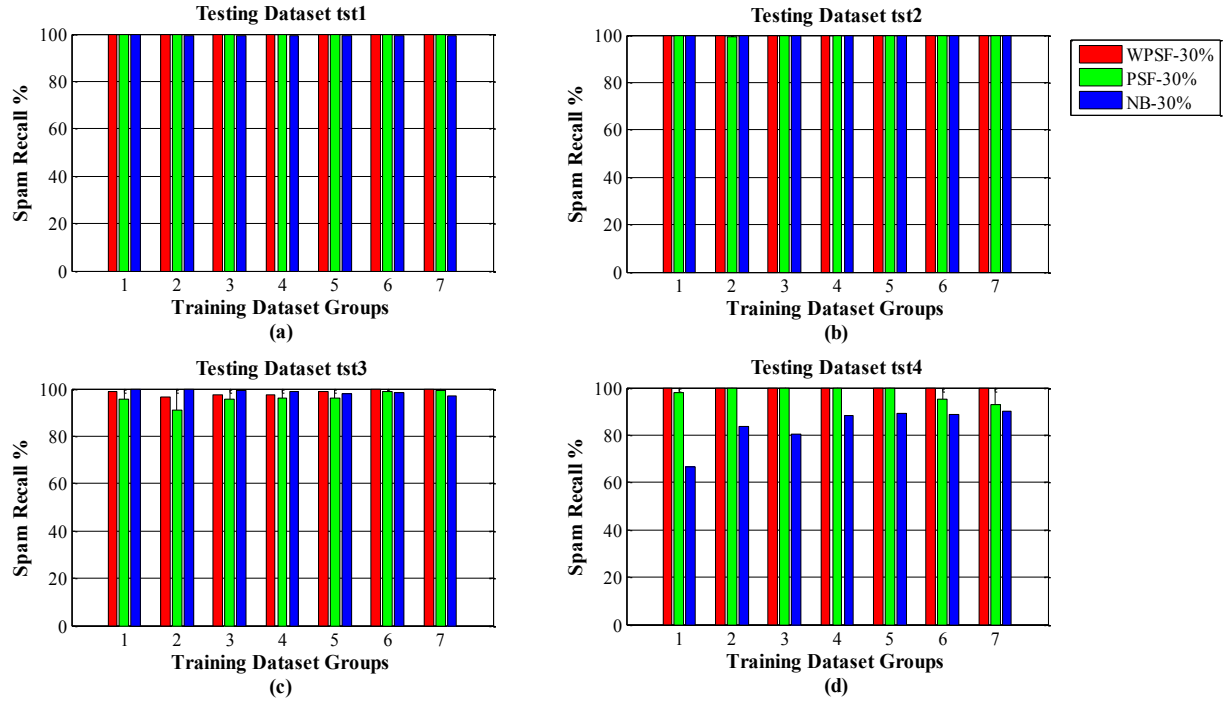

Figure 12- Spam Recall for WPSF, PSF and NB with 17 Features

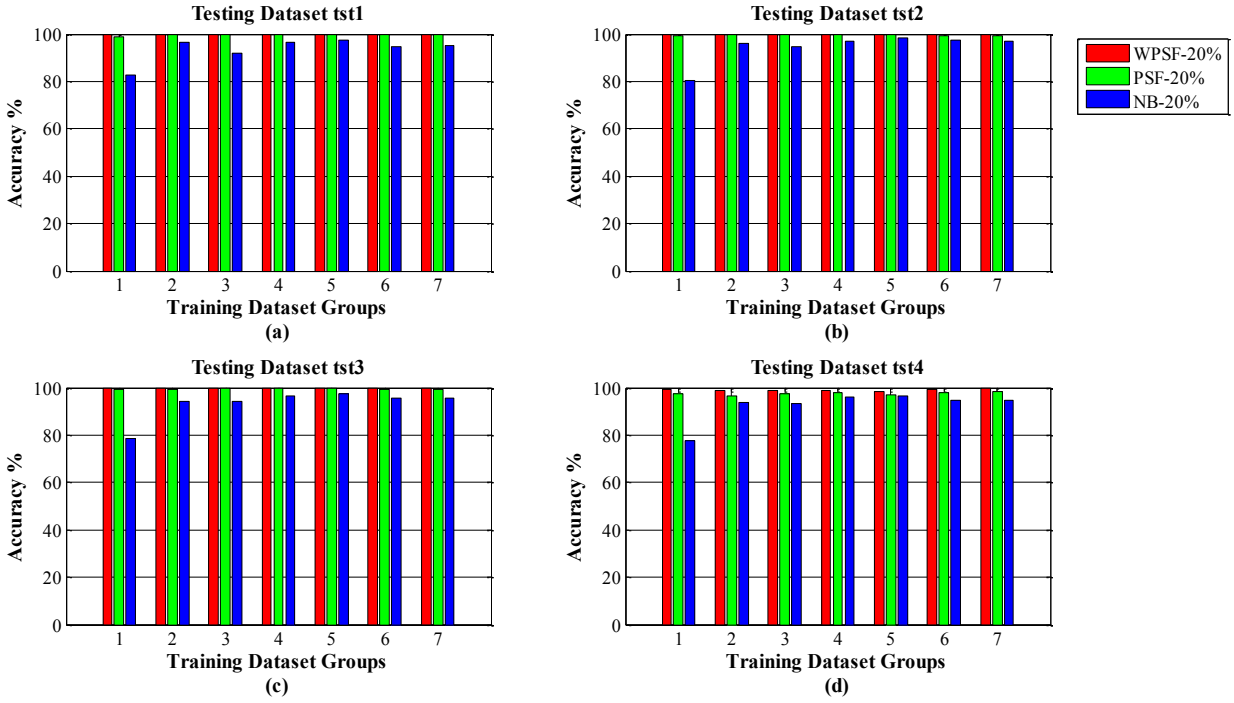

Figure 13- Accuracy for WPSF, PSF, and NB with 13 Features

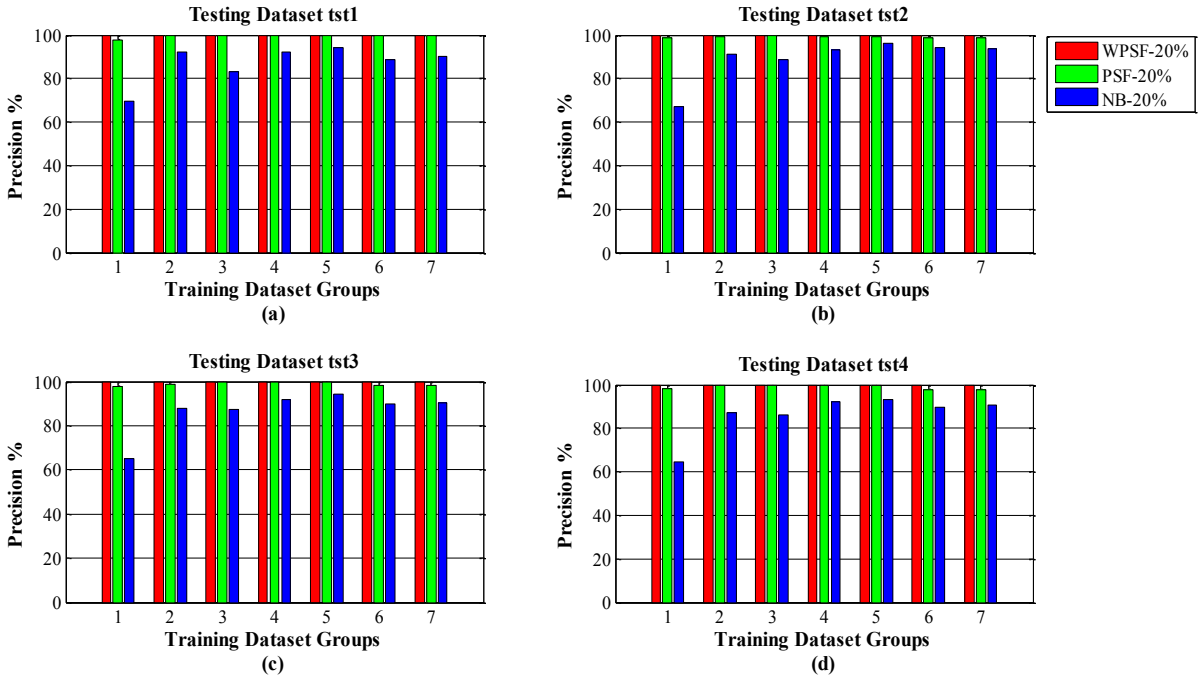

Figure 14- Precision for WPSF, PSF, and NB with 13 Features

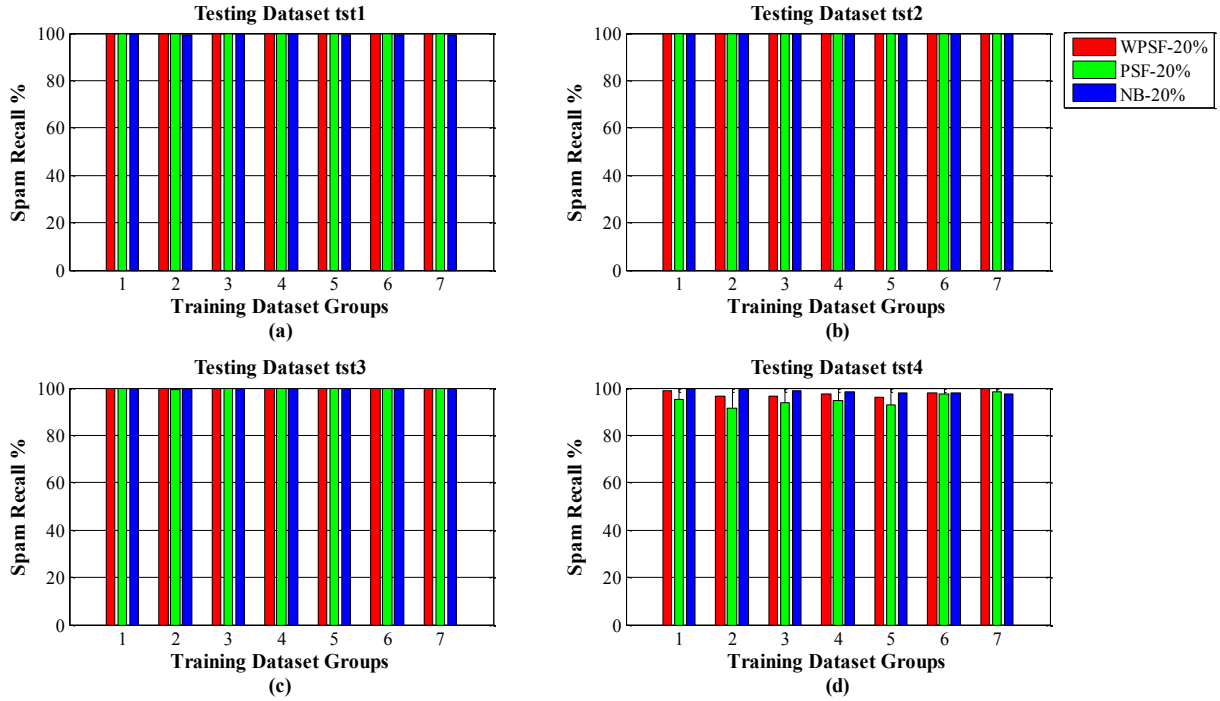

Figure 15-Spam Recall for WPSF, PSF, and NB with 13 Features

6. Conclusions

Spam has become a major problem for companies and private users. This paper proposes a fuzzy based spam filtering technique based on WPCM. Experimental results reveal out that the proposed WPSF model is more efficient than NB and PSF models. In addition, one can see that the size of training dataset has an impact on the effectiveness of the proposed spam filter models. The results reveal that by increasing training dataset size, the performance can also be increased. Moreover, the number of selected features has an effective impact on the performance of WPSF in all evaluation measures.

The performance of WPSF with 20%, 30%, 40% and 50% feature sets in accuracy terms are high and comparable. This is due to that WPSF assigns different weights to different features. Moreover, 40% of feature set gives better accuracy results than using 100% feature set. To this end, one can say that using 40% of feature set can give better compromise between WPSF's accuracy and computation cost.

References

1. El-Alfy, E.-S. M. and Al-Qunaieer, F. S. **2008**. A Fuzzy Similarity Approach for Automated Spam Filtering, International Conference on Computer Systems and Applications, IEEE.

2. Gupta, S. B. and Undavia, J. N. **2012**. A Fuzzy Approach for Spam Mail Detection Integrated with Wordnet Hypernyms Key Term Extraction. *International Journal of Engineering Research and Technology*, **1**(5): 1-5.

3. Yu, Z. Q., Juan, Y. H., Peng W. and Wei, M. **2011**. Fuzzy Clustering based on Semantic Body and its Application in Chinese Spam Filtering. *International Journal of Digital Content Technology and its Applications*, **5**(4): 1-11.

4. Zhong, S., Huang,H. and Pan, L. **2010.** An Effective Spam Filtering Technique Based on Active Feedback and Maximum Entropy, in 7th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2010).

5. Mohammad, N. **2011**. A Fuzzy Clustering Approach to Filter Spam E-Mail, in Proceedings of the World Congress on Engineering, London, U.K.

6. Santhi, G., MariaWenisch, S. and Sengutuvan, P. **2013.** A Content Based Classification of Spam Mails with Fuzzy Word Ranking. *International Journal of Computer Science*, **10**(3): 48-56.

7. Dave, A., Dave, N. and Chauhan, U. **2013**. Text Based Fuzzy Clustering Algorithm to Filter Spam E-mail. *International Journal for Scientific Research and Development*, **1**(3): 635-637.

8. Hameed, S. M. and Mohammed, N. A. J. **2013.** A Content Based Spam Filtering Using Optical Back Propagation Technique. *International Journal of Application or Innovation in Engineering & Management*, **2**(7): 416-421.

9. Elssied, N. O., Ibrahim F. O. and Osman, A. H. **2014.** A Novel Feature Selection Based on One-Way Anova F-Test for E-Mail Spam Classification. *Research Journal of Applied Sciences, Engineering and Technology*, **7**(3): 625-638

10. Hameed, S. M., Mohammed M. B. and Attea, B. A. **2015**. Fuzzy Based Spam Filtering. *Iraqi Journal of Science*, **56**: 506-519.

11. Zhang, Y., Huang, D., Ji M. and Xie, F. **2011**. Image Segmentation Using PSO and PCM with Mahalanobis Distance, **38**(7): 9036–9040, Elsevier.

12. http://archive.ics.uci.edu/ml/datasets/Spambase.