Fuzzy Based Spam Filtering

Sarab M. Hameed*, Marwan B. Mohammed, and Baraa A. Attea
Computer Science Department, College of Science, University of Baghdad, Baghdad, Iraq

Abstract

Emails have proliferated in our ever-increasing communication, collaboration and information sharing. Unfortunately, one of the main abuses lacking complete benefits of this service is email spam (or shortly spam). Spam can easily bewilder system because of its availability and duplication, deceiving solicitations to obtain private information. The research community has shown an increasing interest to set up, adapt, maintain and tune several spam filtering techniques for dealing with emails and identifying spam and exclude it automatically without the interference of the email user. The contribution of this paper is twofold. Firstly, to present how spam filtering methodology can be constructed based on the concept of fuzziness mean, particularly, fuzzy c-means (FCM) algorithm. Secondly, to show how can the performance of the proposed FCM spam filtering approach (coined hence after as FSF) be improved. Experimental results on corpora dataset point out the ability of the proposed FSF when compared with the known Naïve Bayes filtering technique.

Keywords: Cluster prototype, Email, Fuzzy c- means, Information gain, Naïve Bayes, Spam filtering.

تصفية البريد المزعج اعتمادا على الضبابية

سراب مجيد حميد* , مروان بدران محمد , براء علي عطية
قسم علوم الحاسبات ، كلية العلوم، جامعة بغداد، بغداد ، العراق

الخلاصة:

انتشر استخدام البريد الكتروني في عالم الاتصالات والتواصل و مشاركة المعلومات انتشارا متزايد. لكن يعد البريد المزعج واحد من اهم الانتهاكات التي تقلل من فوائد خدمة البريد الكتروني . يمكن للبريد المزعج ان يربك النظام بسهوله لكثرة تكراره ، واغرائته الخداعة لغرض الحصول على معلومات خاصة . يبين هذا البحث اهتمام في تكوين و تكييف وادامة عدة تقنيات لتصفية الرسائل غير المرغوب بها وفرزها تلقائياً دون العودة للمستخدم. المساهمة في هذا البحث ذو جانبين ، الأول هو تقديم منهجية لكيفية تصفية الرسائل غير المرغوب بها على اساس مفهوم الضبابية وخاصة خوارزمية التجميع الضبابي . اما الجانب الثاني فيبين كيفية تحسين اداء الاليه المقترحة لتصفية الرسائل غير المرغوب بها (FSF) . النتائج التجريبية على مجاميع مجموعة البيانات تشير إلى قدرة FSF المقترحة لتصفية الرسائل غير المرغوب مقارنة مع تقنيه التصفية المعروف Naïve Bayes .

*Email: sarab_majeed@yahoo.com

1. Introduction

Nowadays, many places in the internet can be easily overwhelmed with many copies of similar unproductive messages, known as *spam* that costs the sender nearly nothing but, unlike, consumes recipient productivity and system availability. Spam often comes disguise in the form of commercial advertising, mostly for dubious and doubtful products, get rich quick schemes, quasi-legal services, pornography, or viruses [1, 2]. In the last few years, the research community has shown an increasing interest to set up, adapt, maintain and tune spam filtering techniques. The generalized formalization of a spam filtering function, F, is a binary classifier defined , with respect to a set of parameters, Θ, as in Eq. (1):

$$F(m,\Theta) = \begin{cases} C_{spam} \; if \; m \; is \; spam \\ C_{email} \; if \; m \; is \; legitimate \; mail \end{cases} \quad (1)$$

Where m is an electronic message (email) to be classified as either spam or (legitimate) mail. Θ is a vector of parameters that characterizes the spam filtering function, F, and has a great influence on the final classification accuracy. C_{spam} and C_{email} are output labels to be assigned to the received messages.

Usually, spam has unqualified or no absolute definition so as 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 [3], a hybrid spam filtering mechanism based on K-means clustering and support vector machine (SVM) was proposed. The evaluation of the hybrid mechanism was carried out using a spam based standard dataset. The hybridization mechanism results in decreasing training time and increasing accuracy of SVM classifier.

In [4], a fuzzy clustering algorithm (Fuzzy C-Means) was proposed to filter out spam messages. A set of message features, is normalized using Heterogeneous Value Difference Metric (HDVM). Different data set sizes were collected from Spam assassin corpora by real users' emails. They found that the false negative error rate of the algorithm is low between 16% and 4%.

In [5], a text clustering algorithm based on the vector space model is proposed where the features of K-means, Balanced Iterative Reducing and Clustering using Hierarchies BIRCH algorithm were integrated. Nearest neighbor and K-Nearest neighbor clustering (K-NNC) can serve as the basis of classification of the ling spam corpus dataset. Different performance measures such as precision, recall, specificity and accuracy were evaluated. The results concluded that the combination of BIRCH with K-NNC works better with large data sets compared to the performance of K-means.

In [6], 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 this paper, one of the fuzzy clustering family named fuzzy c-means algorithm (FCM) will be utilized 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 Fuzzy c means (FCM). Section 3 illustrates the suggested spam filtering algorithm based on FCM. The proposed algorithm is coined as, FCM spam filtering (FSF). Section 4 illustrates experimental results. Finally, section 5 presents conclusions carried out after this work.

2. Fuzzy c-means Algorithm

Fuzzy c-means (FCM) algorithm is one of the most widely used fuzzy clustering models [7]. The basic concept of FCM algorithm is the generation of the centers of k clusters by contribution of n data points. For each data point or sample, s_i, in a given data set $S = \{s_1, s_2, \dots, s_n\}$, FCM computes its degree of belongingness to each of k clusters. FCM assumes that the number of clusters, k, is known in advance, or at least it is some fixed number. 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 FCM. At the beginning of FCM, the values of these prototypes are chosen randomly. Usually, they are taken randomly from the data set S. Then, according to the Euclidean distance, each sample vector $s_j, 1 \le j \le n$ is assigned a membership (belongingness) degree, $u_{ij} \in [0,1]$, to each cluster v_i. Thus, FCM can construct a

$k \times n$ matrix $U = [u_{ij}]$, the so called fuzzy partition matrix. For fuzzy clustering, the probabilities of belongingness of the data point s_j should be summed up to 1, i.e.:

$\forall_j,\ 1 \leq j \leq n$:

$\sum_{i=1}^{k} u_{ij} = 1$ (2)

FCM algorithm aims to minimize the global cluster variance, i.e., the within cluster variance summed up over all k clusters. This measure is usually denoted by J_m criterion as defined in Eq. 4 [8].

$Min\, J_m(S,U,V) = \sum_{i=1}^{k}\sum_{j=1}^{n} u_{ij}^{m}\, d^2(s_j - v_i)$ (3)

The exponent m in the equation represents fuzziness parameter which is used to adjust the weighting effect of membership values [8].

Since, the objective function $Min\, J_m(S,U,V)$ cannot be minimized directly, an iterative algorithm is used to iteratively optimize the membership degrees and cluster centers by updating $u_{ij}$and v_i using Eq. (4) and Eq. (5) respectively. In other words, first the membership degrees are optimized for fixed prototypes then the cluster prototypes (centers) are optimized for fixed membership degrees.

$$u_{i,j} = \frac{1}{\sum_{i=1}^{k}\left(\frac{d^2(s_j,v_i)}{d^2(s_j,v_k)}\right)^{\frac{2}{m-1}}} \quad (4)$$

$$v_i = \frac{\sum_{k=1}^{n}(u_{i,k})^m S_k}{\sum_{k=1}^{n}(u_{i,k})^m} \quad (5)$$

3. FCM for Spam Filtering

The spam filtering problem can be viewed as a two-class clustering problem, the goal of which is to classify an incoming message into one of two categories; either spam or legitimate email. Due to the vagueness nature of spam, we conduct FCM for this classification purpose. In what follow, we will present the necessary steps of the proposed fuzzy based spam filtering (coined as FSF) together with its mathematical notations and formulations. All notations and formulations will be unified with those presented in the previous sections.

The proposed FSF is mainly composed of two modules: *training module* (FSF_{trn}), and *testing module* (FSF_{tst}). The role of the training module is to train, according to a set $S = \{s_1, s_2, \ldots, s_n\}$ 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.

On the other hand, the goal of the testing module is to make, based on the trained prototype vectors V produced from the training module, a binary classification decision, F, on the incoming message(s). The details of the training and testing module will be described next, after, depicting the general layout of the proposed FSF in figure -1.

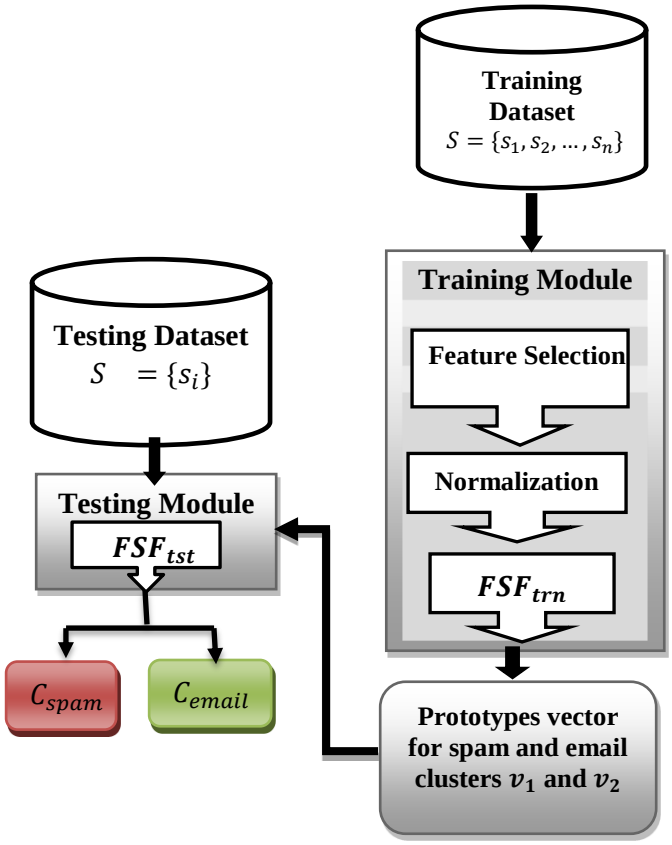

Figure 1- General layout of FSF.

3.1 Training Module

In this module, corpora dataset [9] is used as the input space $S = \{s_1, s_2, \dots, s_n\}$ of messages. Corpora dataset is a widely used spam based dataset created in June/July 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. Table -1 presents the name and type of each attribute exists in the corpora dataset. 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).

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:

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

$s_i = \{s_{i1}, s_{i2}, \dots, s_{i57}\}$ (6)

Table 1: Number, names and types of corpora dataset features.

Feature#	Feature Name	Type
1-48	word_freq_WORD	Continuous (real)
49-54	char_freq_CHAR	Continuous (real)
55	capital_run_length_average	Continuous (real)
56	capital_run_length_longest	Continuous (integer)
57	capital_run_length_total	Continuous (integer)
58	Class label	Nominal

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* [10]. Formally speaking, if $\mathbb{F} = \{\mathbb{F}_1, \mathbb{F}_2, \dots, \mathbb{F}_{57}\}$ is the complete feature set, then, $\mathcal{F} \subseteq \mathbb{F}$. Algorithm 1 outlines the steps necessary for selecting a distinguished feature set.

Algorithm 1 *Feature set selection*

Input:
Corpora dataset: $S = \{s_1, s_2, \dots, s_n\}$
Percentage of Selected feature set: $Per\%$
Output:
Selected feature set: $\mathcal{F} \subseteq \mathbb{F}$
Method:
$S = \{S_1, S_2\}$
S_1={ s_i |s_{i58}=1**}**
S_2={ s_i |s_{i58}=0**}**

// Compute the information need (entropy) for clustering S into two clusters (c_1 and c_2)

$$I(S) = \sum_{i=1}^{2} \frac{|S_i|}{|S|} log_2 \left(\frac{|S_i|}{|S|}\right) \quad (7)$$

For each feature $\mathbb{F}_i \in \mathbb{F}$ $1 \le i \le 57$
Begin
// Divide S into subsets $\{S_1, S_2, \dots, S_l\}$ where S_j is the subset which has the value f_j for feature $\mathbb{F}_i$

//Compute weighted average of information needed to identify the class of an element of each subset

$$E(\mathbb{F}_i) = \sum_{j=1}^{l} \frac{|S_j|}{|S|} I(S_j) \quad (8)$$

//Compute information gain for $\mathbb{F}_i$ by calculating the difference of information before splitting I and information after splitting E.

$$Gain(\mathbb{F}_i) = I(S) - E(\mathbb{F}_i) \quad (9)$$

End
Select *feature set* $\mathcal{F}$ such that $|\mathcal{F}| = Per\%\mathbb{F}$

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]. This can be formulized as:
$\forall \mathcal{F}_i \in \mathcal{F} \mid 1 \le i \le |\mathcal{F}|$

$$\mathcal{F}_i' = \frac{\mathcal{F}_i - \mathcal{F}_i^{min}}{\mathcal{F}_i^{max} - \mathcal{F}_i^{min}} \quad (10)$$

where:
$\mathcal{F}_i$: is the feature value.
$\mathcal{F}_i^{min}$: is the minimum value feature $\mathcal{F}_i$ can get.
$\mathcal{F}_i^{max}$: is the maximum value feature $\mathcal{F}_i$can get.
Now, the input set of samples, S', is ready for handling by the first module of FSF. 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 FCM to define the prototype of each cluster. The main steps of FCM algorithm for FSF is presented in algorithm 2.

Algorithm 2 *FCM for FSF*
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* $|\mathcal{F}|$
- *Number of clusters*$c = 2$.
- *Set fuzziness parameter* m
- *Set iteration number* $t = 0$
- *Stopping criterion* $\varepsilon = 0.001$

Output:

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

Method:

1. At $t = 0$, initialize prototype vectors v_1^0, v_2^0
2. Calculate the Euclidean distance between each sample, s'_j, and the prototype of the two clusters v_1^t, and v_2^t.
$\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\} \wedge \forall k \in \{1, \dots, |\mathcal{F}|\}$

$$d^2(s'_j, v_i) = \sqrt{\sum_{k=1}^{|\mathcal{F}|} (s'_{j,k} - v_{i,k})^2} \quad (11)$$

3. For the two clusters $c1$ and $c2$, and each sample, s_j, compute cluster membership values $u_{i,j}^t$ as:
$\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\}$

$$u_{i,j}^t = \frac{1}{\sum_{k=1}^{2} \left(\frac{d^2(s_j, v_i)}{d^2(s_j, v_k)} \right)^{\frac{2}{m-1}}} \quad (12)$$

4. Update prototype values v_1 and v_2 of the two clusters using:
$\forall i \in \{1,2\}$

$$v_i^{t+1} = \frac{\sum_{k=1}^{n} (u_{i,k})^m s_k}{\sum_{k=1}^{n} (u_{i,k})^m} \quad (13)$$

5. Checking for stopping criteria in Eq. 14, if then stop, else increment iteration number, t, by one and go to step 2.
$\forall i \in \{1,2\} \wedge \forall j \in \{1, \dots, n_t\}$

$$max\{|u_{i,j}^{t+1} - u_{i,j}^t|\} < \varepsilon \quad (14)$$

3.2 Testing Module

In this module, the two prototype vectors calculated in the training module $v_1 = \{v_{11}, v_{12}, \dots, v_{1|\mathcal{F}|}\}$ and $v_2 = \{v_{21}, v_{22}, \dots, v_{2|\mathcal{F}|}\}$ are first extracted. Then, for each of the incoming tested message vectors in a set $S = \{s_i\}$, the two membership values u_{1i} and u_{2i} are computed as follows. The belongingness degree to the spam cluster, u_{1i} is defined in Eq. (15), while the belongingness degree to the legitimate email cluster, u_{2i}, is defined in Eq. (16).

$$u_{1i} = \frac{1}{\sum_{k=1}^{2}\left(\frac{d^2(s_i,v_1)}{d^2(s_i,v_k)}\right)^{\frac{2}{m-1}}} \quad (15)$$

$$u_{2i} = \frac{1}{\sum_{k=1}^{2}\left(\frac{d^2(s_i,v_2)}{d^2(s_i,v_k)}\right)^{\frac{2}{m-1}}} \quad (16)$$

The testing module of FSF, denoted by FSF_{tst}, will assign label C_s or C_e to the tested message s_i . Formulizing FSF_{tst} as in Eq. (1), Θ will represent the prototype set $V = \{v_1, v_2\}$ as follows:

$$FSF_{tst}(s_i, \{v_1, v_2\}) = \begin{cases} C_{spam} \; if \; u_{1i} > u_{2i} \\ C_{email} \; otherwise \end{cases} \quad (17)$$

4. Experimental Results

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

In the training module, a training dataset is divided into seven groups, each contains distinct samples selected randomly from the spam-based dataset. The number of samples for each group is tabulated in table -2.

On the other hand, the testing dataset is divided into four groups, each one contains distinct samples selected randomly from the remaining spam-based dataset. Also, the number of samples for each group is quantified in table -3.

Table 2- Training Dataset Groups

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

Table 3- 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

4.1 Evaluation Criteria

The confusion matrix is used to analyze the effect of the proposed FSF algorithm. As we have two clusters, $c1$ and $c2$, the confusion matrix can be defined as 2×2 matrix. Each entry in the matrix is assigned to one of four possible combinations: True Negative, TN, True Positive, TP, false Positive, FP, and False Negative, FN. TP is considered for spam which is correctly classified, whereas FP occurs when legitimate emails are misclassified as spam. TN considers legitimate messages that are correctly classified, whereas FN occurs when the spam action is misclassified as legitimate email. Table -4 presents the confusion matrix work.

To evaluate the performance of FSF, three criteria are used. These are:

1. **Accuracy (**Acc): this measure reflects the percentage of predictions that are correct [11]. The formula for calculating this measure is given as in Eq. 18.

$$Acc = \frac{TP+TN}{TP+TN+FP+FN} \quad (18)$$

2. **Spam Precision (**SP**):** is the percentage of the predicted positive cases that are correct [11]. The formula for calculating this measure is:

$$SP = \frac{TP}{TP+FP} \quad (19)$$

1. **Spam Recall (**SR**):** is defined as the ratio of the number of correctly detected spam [11]. The formula for calculating this measure is:

$$SR = \frac{TP}{TP+FN} \quad (20)$$

Table 4- Confusion matrix

		Predicted Cluster	
		Negative class (Email)	**Positive class (Spam)**
Actual Cluster	**Email**	TN	FP
	Spam	FN	TP

4.2 Impact of fuzziness Parameter

Increasing and decreasing the value of the fuzziness parameter m has an influence on the performance of FCM. During training phase, m has tested with three setting, 1.5, 2, and 2.5. Table-5 presents the accuracy results of four testing groups while varying m value. Average results presented in table-5

clarifies that best accuracy is achieved when $m = 2$. When $m = 1.5$, average accuracy is 96.5. However, performance of FSF show similar behavior when $m = 2$ or 2.5. We found that, average accuracy results is (98.58%).

Table 5- Impact of fuzziness parameter on FSF's Accuracy.

Training Dataset Groups	m	Testing Dataset Groups				Average
		tst_1	tst_2	tst_3	tst_4	
trn_1	1.5	100	100	98.11	90	97.02
	2	100	99.66	99.66	96.91	99.05
	2.5	100	99.66	99.66	96.91	99.05
trn_2	1.5	100	100	95.77	87	95.69
	2	100	99.66	99.22	95.08	98.49
	2.5	100	99.66	99.22	95.08	98.49
trn_3	1.5	100	100	97.88	90.33	97.05
	2	100	99.83	100	97.33	99.29
	2.5	100	99.83	100	97.33	99.29
trn_4	1.5	100	100	98.55	92.83	97.84
	2	100	99.83	99.88	97.83	99.38
	2.5	100	99.83	99.88	97.83	99.38
trn_5	1.5	100	100	97.88	90.91	97.19
	2	100	99.83	100	97.16	99.24
	2.5	100	99.83	100	97.16	99.24
trn_6	1.5	100	99.83	99.77	95.75	98.83
	2	100	99.83	99.66	98.58	99.51
	2.5	100	99.83	99.66	98.58	99.51
trn_7	1.5	100	99.83	99.88	96.5	99.05
	2	100	99.66	99.44	98.58	99.42
	2.5	100	99.66	99.44	98.58	99.42

4.3 Impact of Number of selected features $|\mathcal{F}|$

Intuitively, increasing or decreasing number of selected features effects on the performance of any spam filtering model, including the proposed FSF. Using information gain, different feature sets can be obtained. Table -6 presents accuracy result of FSF using different percentage of information gain, i.e., different $|\mathcal{F}|$. In the table, the average accuracy of the four testing dataset groups is also included.

Table 6- Impact of Number of features on FSF's Accuracy with m=2

Training Dataset Groups	Percentage	Testing Dataset Groups				Average
		tst_1	tst_2	tst_3	tst_4	
trn_1	100%	100	99.66	99.66	96.91	99.05
	50%	100	99.66	99.22	96.66	98.88
	40%	100	99.66	99.33	96.33	98.83
	30%	100	100	97.22	88.91	96.53
trn_2	100%	100	99.66	99.22	95.08	98.49
	50%	100	99.66	99.33	95.33	98.58
	40%	100	100	96.44	88	96.11
	30%	100	100	95.44	86	95.36
trn_3	100%	100	99.83	100	97.33	99.29
	50%	100	99.83	100	97.25	99.27
	40%	100	99.83	100	96.91	99.18
	30%	100	100	97.66	89.41	96.76
trn_4	100%	100	99.83	99.88	97.83	99.38
	50%	100	99.83	99.88	97.75	99.36
	40%	100	99.83	99.88	97.83	99.38
	30%	100	100	98.44	91.75	97.54
trn_5	100%	100	99.83	100	97.16	99.24
	50%	100	99.83	99.88	97.33	99.26
	40%	100	99.83	100	97	99.20
	30%	100	100	97.55	88.75	96.57
trn_6	100%	100	99.83	99.66	98.58	99.51
	50%	100	99.66	99.55	98.08	99.32
	40%	100	99.83	99.55	98	99.34
	30%	100	99.83	99.44	98	99.31
trn_7	100%	100	99.66	99.44	98.58	99.42
	50%	100	99.66	99.11	98.5	99.31
	40%	99.66	99.66	99.33	98.41	99.26
	30%	99.66	99.5	99.33	98.08	99.14

In this experiment, fuzziness parameter m is set (according to table - 5) to 2. Four 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) and 30% of information gain (i.e. 17 features are selected). The results in table - 6 indicate that even using the whole feature set, i.e., $|\mathcal{F}| = 57$ results in the highest accuracy, but discarding about half of the total feature set (i.e., letting $|\mathcal{F}| = 29$) can also produces comparable results. Moreover, in two cases (trn_2 and trn_5) out of the seven cases mentioned in the table, 50% of information gain gives better accuracy results than using 100% feature set. To this end, one can say that using 50% of feature set can give better compromise between FSF's accuracy and computation cost.

Comparative Results

The following section illustrates the performance of FSF and Naïve Bayes (NB) in terms of *accuracy*, *spam precision* and *spam recall*.

Figure -2 depicts the Accuracy of FSF and NB with 50% of features) when seven training dataset groups and four testing dataset groups are used**.** The results reveal that FSF has higher accuracy than NB regardless of number of samples in training and testing dataset groups.

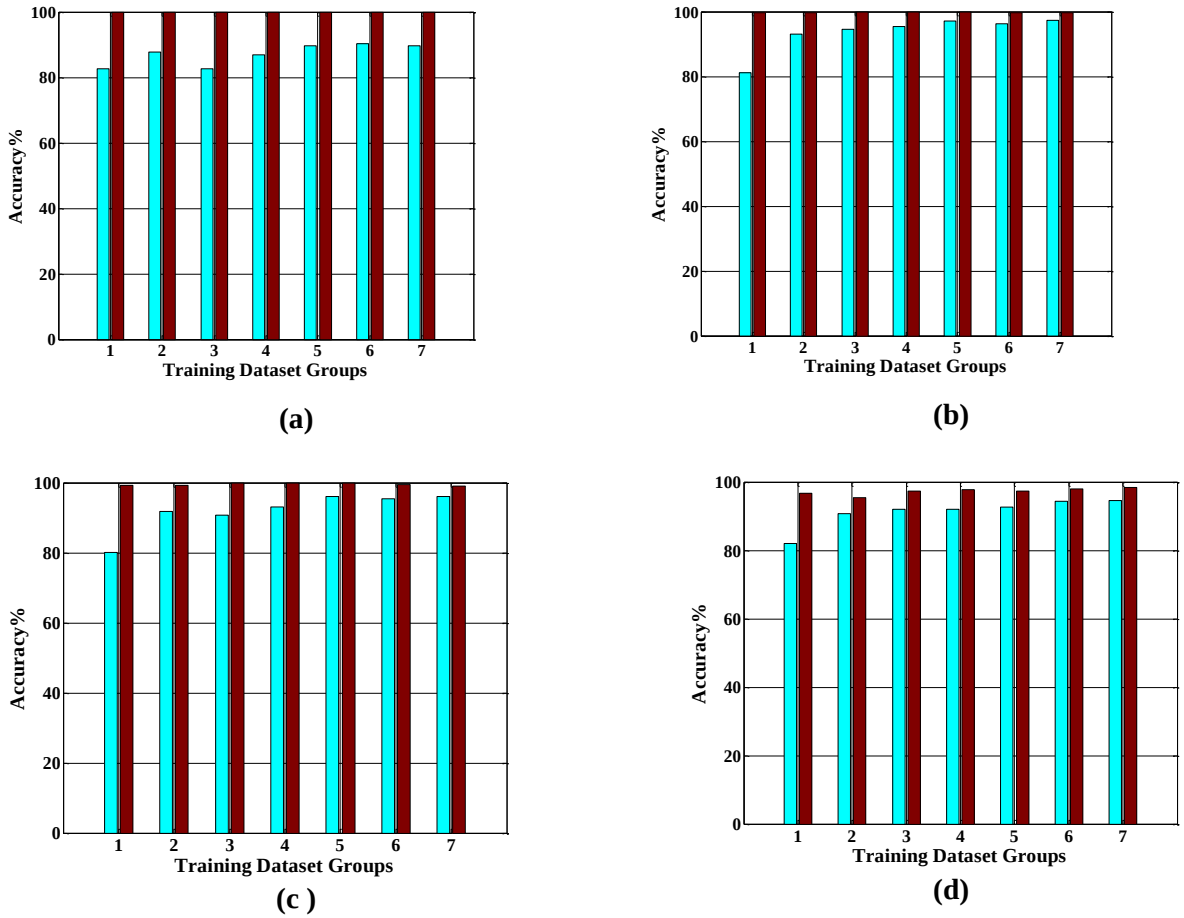

Figure 2- Accuracy while using each of the seven training datasets $\{trn_1, trn_2, \dots, trn_7\}$ for testing (a): tst_1 dataset (b): tst_2 dataset (c): tst_3 dataset (d): tst_4 dataset. In each figure, NB (left bar) and FSF (right bar).

Figure-3 depict the precision of FSF and NB with 50% of features) when seven training dataset groups and four testing dataset groups are used**.** The results clarifies that FSF provides higher precision than NB in all training and testing groups.

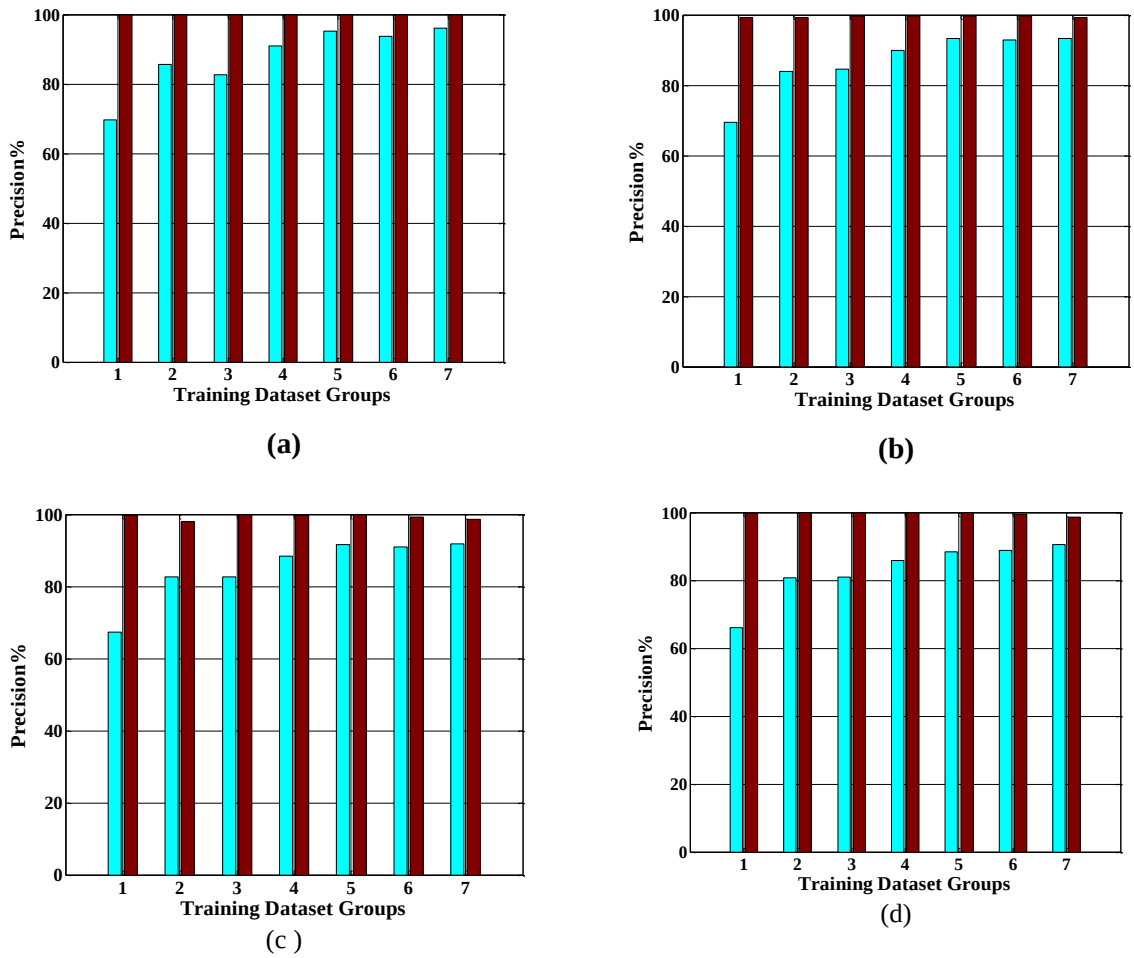

Figure 3- Precision while using each of the seven training datasets $\{trn_1, trn_2, \dots, trn_7\}$ for testing (a): tst_1 dataset (b): tst_2 dataset (c): tst_3 dataset (d): tst_4 dataset. In each figure, NB (left bar) and FSF (right bar).

Figure- 4 depicts the recall of FSF and NB with 50% of features**.** The spam recall of FSF and NB are equal for testing groups $\boldsymbol{tst_1}$ and $\boldsymbol{tst_2}$. On the remaining two testing datasets, NB's spam recall is better than or equal to spam recall of FSF. 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 FSF tends to maximize both accuracy and precision at the expense of spam recall function.

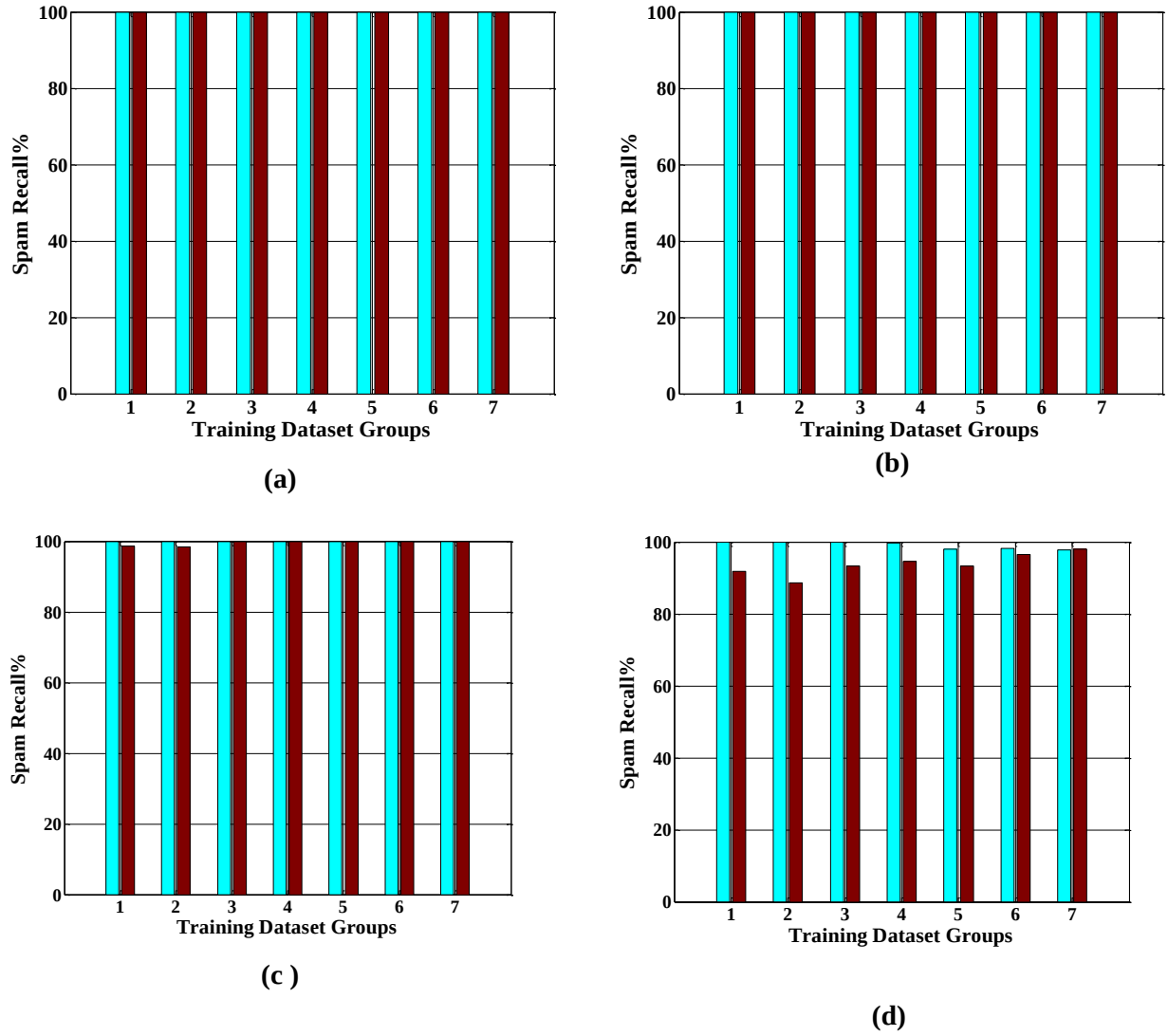

Figure 4- Spam recall while using each of the seven training datasets $\{trn_1, trn_2, \dots, trn_7\}$ for testing (a): tst_1 dataset (b): tst_2 dataset (c): tst_3 dataset (d): tst_4 dataset. In each figure, NB (left bar) and FSF (right bar).

5. Conclusions

Spam has become a major problem for companies and private users. This paper proposes a fuzzy based spam filtering technique based on FCM. Experimental results reveal out that the proposed FSF algorithm is more efficient than Naive Bayes algorithm. While FSF achieves its highest accuracy result in (98.58%), the result of Naïve Bayes is (96.44%). Furthermore, When information gain algorithm is applied (e.g., when selecting only 50% of features) the results of FSF is better than NB However, the spam recall of NB is better than or equal to spam recall of FSF. Further extension to the current work can be recommended. For example, trying to use another fuzzy clustering algorithm, such as possiblistic fuzzy c-means.

References

1. Yeganeh, M. S., Bin L. and Babu, G. P. **2012.** A Model for Fuzzy Logic Based Machine Learning Approach for Spam Filtering.*IOSR J. Computer Engineering*, (4), pp: 07-10.

2. Suryavanshi, A. and Shandilya, S. **2012.** Spam Filtering and Removing Spam Content from Massage by Using Naive Bayesian. *International J. Computational Engineering & Management*, (15).

3. Elssied, N. O. F., Ibrahim,O. and Ublbeh, W. **2014.** Unimproved Of Spam E-mail Classification Mechanism Using K-Means Clustering. *J. Theoretical and Applied Information Technology,* 60 (3).
4. Mohammad, N. T. **2011**. A Fuzzy Clustering Approach to Filter Spam E-Mail , Proceedings of the World Congress on Engineering, Vo. III l, London, U. K.
5. Basavaraju, M., and Prabhakar, R. **2010**. A Novel Method of Spam Mail Detection using Text Based Clustering Approach. *International J. Computer Applications*, (5), pp: 0975 – 8887.
6. Hameed, S. M. and Mohammed, N. A. J. **2013**. A Content based Spam Filtering Using Optical Back Propagation Technique, *International J. Application or Innovation in Engineering & Management* (IJAIEM), (2).
7. Pal, N. R., Pal, K. Keller, J. M., and Bezdek J. C. **2005**. A Possibilistic Fuzzy c-Means Clustering Algorithm, *IEEE Transactions on Fuzzy Systems,* (13).
8. Yafan, Y., Dayou, Z., and Lei, H. **2008**. Improving Fuzzy C-means Clustering by a Novel Feature-weight Learning, *IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application.*
9. http://archive.ics.uci.edu/ml/datasets/Spambase.
10. Fu X, Liu L., Gong T. and Tao L. **2011**. Improving Text Classification with Concept Index Terms and Expansion Terms. *Advances in Neural Networks,* (6677), pp: 485-492.
11. Delany, S. J., Cunningham, P, and Coyle L. **2005.** An Assessment of Case-Based Reasoning for Spam Filtering. *Artificial Intelligence Review,24(3-4)* pp: 359-378, Springer Verlag.