A Genetic Based Approach to Optimize The Fuzzy Clustering Spam Filters

Nehaya.T.Mohammad

Abstract - Spam email, is the practice of frequently sending unwanted email messages, usually with commercial content, in large quantities to a set of indiscriminate email accounts. Effort has been put into solving the spam problem from many directions. We examine the use of an optimizing technique to detect the best value of the Fuzzy Clustering Parameters which are the number of clusters and the Fuzzifier value are experimentally set and have a noticeable influence on the success classification rate of the algorithm and considered in our Fuzzy Clustering spam classification as a spam fighting technique. Our results show that the Genetic algorithm can improve the performance of the Fuzzy C-mean algorithm. The optimized algorithm scores a total success rate of 94.9% on the tested data.

Index Terms- Spam Filtering, Genetic Algorithm, Fuzzy Clustering

I. INTRODUCTION

Spam, or unwanted commercial email, has become an increasing problem in recent years. Estimates suggest that perhaps 70% of all email traffic is spam. As spam clutters inboxes, time and effort must be devoted to either deleting it after it is received, or preventing it from even reaching the user [9]. The problem of spam multiplies daily, and is an annoyance to every user of email. Some estimates suggest that the average per person is 10 working days per year spent solely dealing with spam [10].

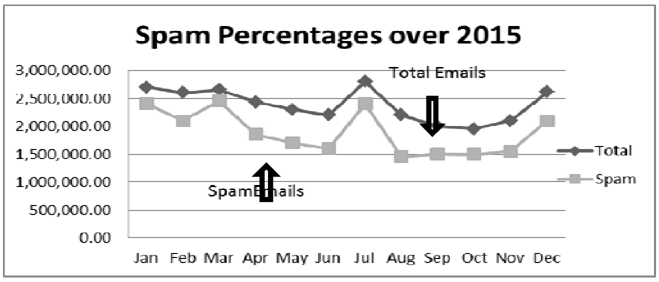

Fig.1. Spam as a percentage of all email traffic on 2015

As Shown in Figure 1[18]. The proportion of spam to Legitimate email changes over time. Throughout 2015, spam comprised around 81-87 percent of all email traffic observed by App River's filters, which puts the spam filtering in the category of skewed class distribution problems [6].

Most of the research in spam filtering concentrates on using data mining approaches to solve this problem [6]. Generally by treating the spam filter as a static text classification problem . According to data mining, the spam is a classification problem where the filtering system aims at distinguishing spam from legitimate emails. Thus, classification algorithms that are widely used for pattern recognition can strongly be used to solve the spam problem. A misclassified spam that arrives in a user's inbox is annoying. A misclassified ham that the user never sees may result in loss of business, productivity, opportunity, or time. Spammers actively attempt to defeat spam filters by substituting look-alike characters for letters, hiding random text in an email, misspelling words, including pictures that
Show the advertisement, or embedding links into deceptively-phrased emails.

Therefore any anti-spam technology must be able to adapt quickly. Automated methods of spam filtering that can learn how to distinguish spam emails from ham emails and can be trained – learn in an updatable fashion - are of vital importance. A good anti-spam technique will have three characteristics: it will accurately classify spam and ham, it will be easily adaptable, and it will be easily scalable [17].

Most of the current research in spam filtering concentrates on using data mining approaches to solve the spam filtering. According to data mining, the spam is a classification problem where the filtering system aims at distinguishing spam from legitimate (ham) emails. Thus, classification algorithms that are widely used for pattern recognition can be used to solve the spam problem.

The spam filter success rates are varied through several approaches, until the moment, the intend to optimize spam filter is a new idea, as long as there was one optimal solution to the spam problem currently, with thousands of spammers looking for new ways to defeat it, the payoff of research into alternate methods is apparent[17].

Nehaya .T.Mohammad is with Department of Computer Information Systems, University of Jordan. (Email: nehaya.tayseer@gmail.com)

In this paper we investigated the effectiveness of several spam filtering techniques and technologies. The filtering is done by using Fuzzy Clustering algorithm optimized by the genetic algorithm and the results are compared with the Bayesian, NN, and SVM Classifiers. Our analysis was performed by simulating email traffic under different conditions. Our demonstration supported that the genetic algorithm based spam filters results are promising in comparison with the other classifiers. The Classification accuracy obtained a rate above 94% and the low false positive rates are achieved in many test cases.

The rest of the paper is organized as follows. Section 2 overviews spam filtering work that look at common methods of fighting spam, including artificial intelligence influenced techniques. The Genetic-fuzzy spam filtering optimized technique is given in section 3. Experiments and results are illustrated in section 4. Finally, conclusions and future work are given in section 5.

II. RELATED WORK

Various techniques exist for filtering spam. These methods can be generally categorized into techniques that have been influenced by artificial intelligence and machine learning, and other techniques. These other techniques tend to be older and less robust. For example, use of white lists, black lists, and gray lists is straightforward; if the email is sent from a known spammer, it is marked as spam; if it is sent from a user-approved address, it is allowed through to the inbox. Anything else is "gray listed" to a folder where the user can approve it as valid or mark it as spam. The difficulty with this approach is that the burden on the user can be considerable. Rules-based spam filters apply pre-written rules to a spam, such as "if subject contains 'Viagra', email is spam". These may accidentally result in misclassification of a real email as spam, classification of spam as valid email, and must be updated frequently to stay abreast of spammers' techniques. Both of these techniques have their place; however, they should not be relied upon as the only filter. Content-based filters are founded on the premise that it is possible to create a set of rules, exemplars or features that represent the degree to which an email is to be considered as a spam, and that if this is over some threshold, is considered to be spam. Such filters have been the focus of considerable interest, with work on rule-based filters, nearest neighbor classifiers [12], decision trees [5] and Bayesian classifiers [11]. Initial implementations of these filters were centralized, but with spam comprising 50% of all emails traffic. As the knowledge base is now in the hands of the system Administrators, it can be customized to suit the characteristic email and spam that individual domains receive. Users can feed information back about false positives and false negatives that enables the filter to be retrained. Spam Assassin given in [13] is perhaps the most known example of this approach. Thus the huge content-based filters have been developed towards a higher degree of collaboration as they have become decentralized Clutters. Machine learning techniques are more varied and flexible [17].

Decision trees [5] classify email as spam or ham based on previous data. They are costly to calculate and recalculate as spammers change techniques. Bayesian networks [11] are the most popular anti-spam technique currently, but they can be difficult to scale up and rely on many features to make their judgments.

III. OPTIMZED FUZZY SPAM FILTER MODEL

In Fiqure2, we propose our Genetic based FCM Filter. The model has three main phases; in the first phase, the proposed model is mainly learned using a set of training data. In the training phase, the FCM algorithm parameters Values (Fuzzifier and the number of clusters) are randomly set. In the second stage, the trained algorithm is tested on a different testing email sets. The testing phase is performed many times until a good success rate is reached; in our case the accepted success rate was 88%. The GA Optimizer is executed as a last stage which will be applied to obtain the optimal success classification rate by getting the best values of the FCM Parameters.

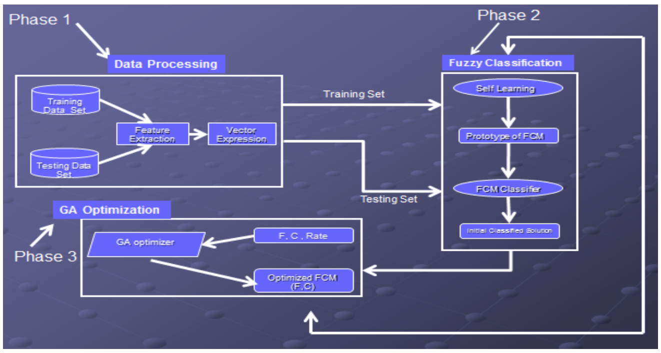

Fig.2 the Genetic Fuzzy Clustering Spam Filter Model

Our Experimental results shows that the classification Rates obtained are encouraging, hence the proposed approach can be effectively used as optimizer.

A. Data Processing phase

At this stage the Fuzzy clustering Identified features were extracted from the input data set. These features were previously identified in our published work FCM [17].

B. Training phase (Building Up prototype file)

At this stage, a prototype was created which represents the mean and the standard of all the features; also the clusters identified by the centers are generated from the FCM algorithm. Each cluster contains multiple of spam and ham emails generated by randomize technique at the first run. Assuring that no clusters have the same number of spam s and hams. We have applied the FCM and distance metric algorithms which modified the randomize weights of each email by defining the best cluster to contain its relevant emails by picking up the shortest distance between the email vector and the cluster identified, this is clearly giving us a high level explanation of the fuzzy clustering working mechanism.

C. Testing Phase

The Testing phase is also considered in phase 2 where the FCM and DM algorithms were performed consequently and come up with the tuned prototype.

D. Optimization Phase

With the optimization stage, the genetic algorithm intends to optimize the Classification process of the Spam/ham emails. Brief details about the mechanism of the Genetic algorithm will be explained as the starting point of an evolutionary process by the Genetic Algorithms is a random population. The population consists of a set of solutions for the problem (called chromosomes), each solution is evaluated by a fitness function value “f”, and the fitness function measures how well the solution can solve its task.

When the fitness function is calculated for each member of a population, a new generation is formed. This means that a new population is generated from the old ones. In our case the fitness function is calculated and equals to the error rate for each prototype that is generated by the training stage in the FCM model.

There are three different techniques used to specify the solutions in the GA population; the first technique is to do the selection of the members process from the old population, the second technique is to do the crossover that combines the two solutions from the old population to generate the new population, where the third technique is to do the mutation which changes some random aspect of a certain fraction of the new population [19].

In our case we used the second technique , the crossover process is executed multiple times until at least one of the solution’s fitness function values exceeds a threshold that has been pre-determined which equals to Epsilon or until reaches the identified iterations which equals to 30 iterations. Figure 3 shows the detailed flowchart of our GA optimizer which summarized by the following steps:

1. Generate the population of FCM solutions which picked randomly.
2. Determine the Fitness function using the formula defined in the GA

Where the fitness Function f is equal to the Error Rate

$$P(s_i) = \frac{f(s_i)}{\sum_{s_{j\in p}} f(sj)}$$

the selection process was performed by selecting the initial population from the sorted array which is sorted according to the error rate in ascending order, our proposed solution takes the lowest error rate of the two parents and then do perform the crossover to produce the next population, this process will go into several iteration until we have got the optimal value of the fuzzifier and clusters numbers which give the higher classification success rate. We stopped our iterations with a success rate equals to 94% and reaching stopping criteria of error rate difference equals to .05.

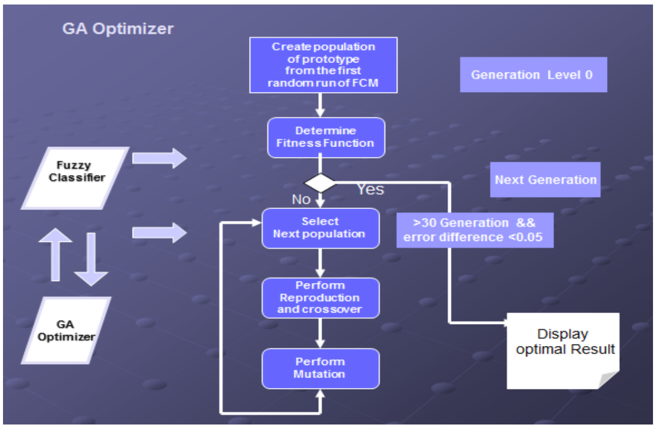
GA Optimizer
Create population of prototype from the first random run of FCM
Generation Level 0
Determine Fitness Function
Fuzzy Classifier
Next Generation
Yes
No
Select Next population
>30 Generation && error difference <0.05
GA Optimizer
Perform Reproduction and crossover
Display optimal Result
Perform Mutation

Fig.3 Flowchart of GA Optimize

IV. EXPERIEMNTS AND RESULTS

The accuracy of a spam filter is obtained by testing the filter on a selected testing set. However most of the testing done is based on existing e-mail corpora. Some of the corpora have been reported to account for a million of mixed legitimate and spam e-mail messages. Part of the corpus is used to train the spam filter and the remaining is used to verify the accuracy using mixed portions of training to testing ratios.

We have conducted two experiments to evaluate the efficiency of the Genetic based Fuzzy clustering proposed filter, our objectives are respectively to demonstrate the performance of Fuzzy clustering under the situation of applying genetic optimization algorithm on a fixed data set and to compare the efficiency of the proposed approach with reference to other machine leanings approach used for spam filtering.

Experiment I

In experiment I, we demonstrate the performance of Fuzzy clustering using the Genetic algorithm on a fixed data set.
As shown in Table 1, the GA Parameters are set in terms of the dependency with the FCM parameters only in the initial population phase, and then it will reproduce new generation through the crossover process and optimize the value according to the fitness function.

TABLE I: SETTING FOR EXPERIMENT I

FCM Parameters	**Number of Clusters**	**Set Randomly**
	Fuzzier	Set Randomly
	Distance Metric	HVDM
	Initial Setting of weights	Random
	Stopping criteria	max change < 0.005
	Normalization	Yes
GA	Initial Parameters	Fuzzifier, Clusters,
	Stopping criteria	Iteration which bounded between 10 to 30 by iterations.
	Fitness function	Error Rate
	Number of solutions	10 to 20
Training Set and Testing Set	Spam Proportion	50%
	Ham Proportion	50%
	Size	12000
Training Set	Ratio	20%,30,
Testing Set	Ratio	80%,70%

The results of applying the GA are shown in Table 2; we considered the baseline experiments as mentioned by Chih-Chin Lai [7]. Using the same data set ratio, and considering the email subject and the body message, we have compared our resulted rates with the results of difference approaches as listed in the table. our GA-FCM approach Produce higher success rate equals to 91.44% which is greater than the obtained rates for Naïve Bayesian Filter (NB), and K-nearest neighbor Filters (KNN), given that we have used the sample set of 30% data used for training and 70% data used for testing.

TABEL II RESULTS OF EXPERIMENT I

	Methods									
	NB		**KNN**		**SVM**		**FCM**		**Genetic_ FCM**	
Emails txt Ratio	*Training: Testing*		*Training : Testing*		*Training : Testing*		*Training : Testing*		*Training: Testing*	
	20:80	30:70	20:80	30:70	20:80	30:70	20:80	30:70	20:80	30:70
Succe ss Rate	86.67	85.23	40.22	40.75	92.24	90.13	67	83.9	91.44	90.08

Experiment II

In this experiment, we have streamlined the proposed approach using different training and testing data set to perform a thorough comparison with reference to other spam filters approaches [16].Table3 shows the set up values made. We used a stopping criteria in the GA Optimization process which underlined the difference in the value of the error rate on each iteration to be close to .05[3].

TABLE III SETTING FOR EXPERIMENT II

FCM Parameters	**Number of Clusters**	**Set Randomly**
	Fuzzifier	*Set Randomly*
	Distance Metric	*HVDM*
	Initial Setting of weights	*Random*
	Stopping criteria	*max change < 0.005*
	Normalization	*Yes*
GA	*Initial Parameters*	*Fuzzifier, Clusters*
	Stopping criteria	*Rate change <.05 from the top rated solution of each iteration and the iterations is bounded between 10 to 30 by iterations .*
	Fitness function	*Error Rate*
Training Set and Testing Set	*Spam Proportion*	*37%*
	Ham Proportion	*63%*
	Size	*6000*
Training Set	*Ratio*	*20%,30%,40%,50%*
Testing Set	*Ratio*	*80%,70%,60%,50%*

Figure4 shows the approaches were used for the spam classification defining four different ratio of the training: testing sets [16], the first two ratios (20:80) & (30:70) respectively, our approach gives more enhanced success rate than NN ,the reason can be concluded here that the NN classifier is more sensitive to the change of the training set size as its parameters are highly dependent on this set, more over the generalization ability is poor in the NN model [7], for these reasons the NN is not suitable to be used alone as spam detection tool. In comparison with NB approach, our success rate is potentially close to the rates of the NB on the first two sample sets. On the other hand, given the second part of the data sets 3&4 with a ratios of (40:60) &(50:50), our approach has achieved a success rate of 93.4% and 94.9% which is potentially the higher success rate score and almost reaches the classification rate of the SVM (Support Vector Machine). Our stopping condition determined a value of the fuzzifier equals to 1.53 and the number of clusters equals to 9 clusters giving an error rate of the submitted solution around to .056 which are the best value to obtain the above mentioned classification success rate.

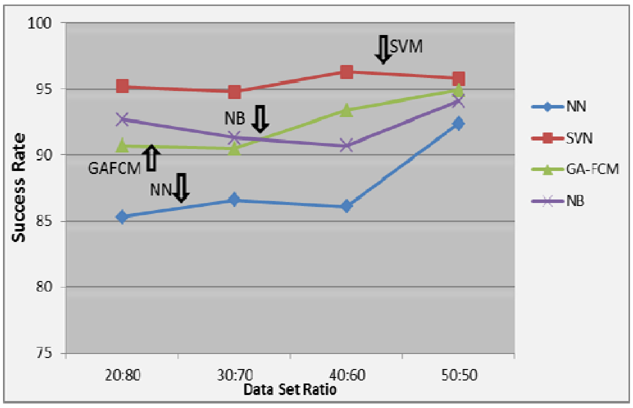

Fig 4: Results for Experiment II.

Our concluded work supported the use of the Genetic algorithm as a sufficient optimizer. In figure 5, we have presented the four data ratio we used in our proposed Spam Classification Model using the GA Optimizer along with the Classification success rate. The more the training emails data set is taken, the more high success is obtained. Nevertheless, a potentially accepted success rate is also obtained with the smallest data set size, which concludes that the GA is not sensitive approach model. As an optimizer, it depends on the solutions size, stopping criteria, and the cross over process rather than the emails set size.

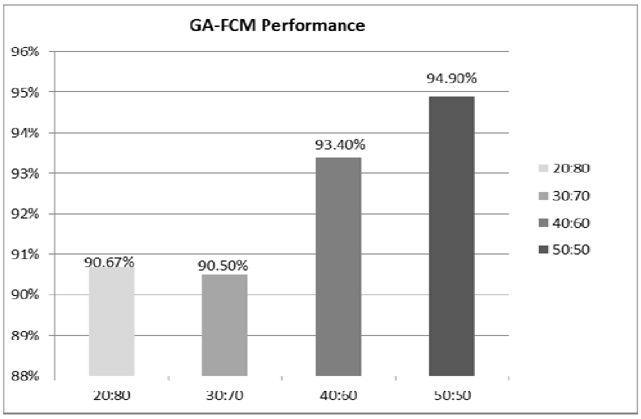

Fig 5: GA Performance over different data sets ratio

V. CONCLUSIONS AND FUTURE WORK

Spam Filtering is a problem of great importance and has gained a great attention in the last decade. The problem's difficulty and interestingness arises from the changing nature of spam. The high accuracy required from any useful spam filter makes the problem even more demanding. In this paper, the Genetic Fuzzy C-Mean Clustering algorithm is evaluated as a tool of building an optimized spam filter. The algorithm was tested with a set of features normalized using the HDVM functions. The approach has been testing using a variant proportion of spam emails which reflects nature of the problem. The approach is evaluated using a standard model suggested by [6] for evaluating spam filters.

The results were promising. The false positive error rate did not exceed 1.5% and stabled around 0.7% when ham emails proportion is more than 50%. We achieved between 10% to 4% for the false negative error rate. These results support our hypothesis regarding the suitability of the combined approaches used. Our approach takes advantage of the generalization ability of the FCM algorithm, extracts representative features from the data, and uses a suitable distance metric.

Finally, our future work includes optimizing more Parameters; which are the feature selection process in order to lower the false positive rates and testing other source of data sets.

REFERENCES

[1] S.Aksoy,and R.M. Haralick,"Feature normalization and likelihood-based similarity measures for image retrieval". Pattern Recognition Letters, 22(5):563—582., 2001

[2] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, and C. D. Spyropoulos, "An experimental comparison of naive bayesian and keyword-based anti-spam Filtering with personal e-mail messages", In Proc. Of SIGIR-2000, ACM, 2000.

[3] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum Press, New York, 1981.

[4] Hovold, J. (2005). "Naive Bayes spam filtering using word-position-based attributes and length-sensitive classification thresholds". Proc. of the 15th NODALIDA conference, Joensuu, 84-93.

[5] Carreras, X., and Marquez, L., 2000, "Boosting trees for anti-spam email filtering", Proc. of SIGIR-2000, ACM, 160-167.

[6] T. Fawcett, "In vivo" spam filtering: a challenge problem for KDD, ACM SIGKDD Explorations Newsletter, 5(2): 141-148, 2003.

[7] Lai, Chih-Chin. (April-2007). "An empirical study of three machine learning methods for spam filtering". Knowledge-Based Systems, Volume 20, Issue 3, pp.249-254

[8] Zhao, W. And Zhang, Z. (2005). "An e-mail classification model based on rough set theory". In Proc. of the Int. Conf. on Active Medial Technology, pp. 1-6.

[9] Message Labs Spam Intercepts data, 2006,http://www.messagelabs.com/published lish/threat_watch_dotcom_en/threat_statistics/spam_intercepts/DA_114633.chp.html.

[10] N. Nie, A. Simpser, I. Stepanikova, and L. Zheng, "Ten years after the birth of the internet, how do Americans use the internet in their daily lives?" Technical report, Stanford University, 2004.

[11] M. Sahami, S. Dumasi, D. Heckerman, and E. Horvitz, "Abayesian approach to filtering junk e-mail. In Learning for Text Categorization", Papers from the 1998 Workshop, Madison, Wisconsin, 1998.

[12] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. Spyropoulos, and C. Stamatopoulos, "A Memory-Based Approach to Anti-Spam Filtering for Mailing Lists", Information Retrieval, 6:49–73, 2003.

[13] SpamAssassinPublicCorpus,2006, http://spamassassin.apache.org/publiccorpus/

[14] E. P. Xing, A. Y. Ng, M. I. Jordan, and S. Russell, "Distance metric learning, with application to clustering with side-information", In Advances in Neural Information Processing Systems 15, pages 505--512, Cambridge, MA, 2003. MIT Press.

[15] S. Nascimento, "Fuzzy Clustering Via Proportional Membership Model", IOS Press, 2005

[16] Bo Yu, Zong-ben Xu, (4, May 2008). "A comparative study for content-based dynamic spam classification using four machine learning algorithms". Knowledge-Based Systems, Volume 21, Issue, pp. 355-362.

[17] N.T.Mohammad" A Fuzzy Clustering Approach to Filter Spam E-Mail "- ISBN: 978-988-19251-5-2

[18] Apprivers Spam filter https://www.appriver.com/about-us/security-reports/global-security-report-end-of-year-2015/

[19] Garcı´a,R. Oliveira,M. Maldonado,J. (2005). " Genetic algorithms to support software engineering experimentation". Proceedings of the International Symposium on Empirical Software Engineering, IEEE Computer Society, Noosa Heads, Australia, pp. 488–497.