SF-HME system: A Hierarchical Mixtures-of–Experts classification system for spam filtering

Petros Belsis
Department of Information and Communication Systems Engineering
University of the Aegean
Karlovasi, Samos, Greece
pbelsis@aegean.gr

Kostas Fragos
Department of Electrical and Computer Engineering
National Technical University of Athens
Zografou, Athens, Greece
kfragos@ece.ntua.gr

Stefanos Gritzalis
Department of Information and Communication Systems Engineering
University of the Aegean
Karlovasi, Samos, Greece
sgritz@aegean.gr

Christos Skourlas
Department of Informatics
Technological Education Institute of Athens,
Aigaleo, Athens, Greece
cskourlas@teiath.gr

ABSTRACT

Many linear statistical models have been lately proposed in text classification related literature and evaluated against the Unsolicited Bulk Email filtering problem. Despite their popularity - due both to their simplicity and relative ease of interpretation - the non-linearity assumption of data samples is inappropriate in practice, due to its inability to capture the apparent non-linear relationships, which characterize these samples. In this paper, we propose the SF-HME, a Hierarchical Mixture-of-Experts system, attempting to overcome limitations common to other machine-learning based approaches when applied to spam mail classification. By reducing the dimensionality of data through the usage of the effective Simba algorithm for feature selection, we evaluated our SF-HME system with a publicly available corpus of emails, with very high similarity between legitimate and bulk email – and thus low discriminative potential - where the traditional rule based filtering approaches achieve considerable lower degrees of precision. As a result, we confirm the domination of our SF-HME method against other machine learning approaches, which appeared to present lesser degree of recall.

Keywords

Spam mail, Machine learning, Hierarchical systems of Experts

1. INTRODUCTION

Email has become lately the dominant way of remote communication. The cost of email is virtually zero comparing to traditional massive marketing notification techniques [16] [17], making it an attractive way to unethical companies to communicate with potential customers. Unfortunately, the emergence of this extremely useful means of communication did not come without its drawbacks, due to the fact that it is prone to malicious users. Unsolicited Bulk Email or most commonly *spam mail*, produces considerable problems to Internet Service Providers and becomes annoying to common Internet users that are obliged to spend considerable amount of time distinguishing the legitimate from the spam mails.

Several solutions have been proposed towards the alleviation of the problem, from technical to regulatory and economic [20]. Filtering is among several popular technical solutions [18] [19]. Several commercial or open source mail clients offer filtering capabilities to the average user, while other, server side mail processing products require manual configuration and constant update by administrators. These approaches are distinguished by their high cost and administrator's personal commitment as well as for their ineffectiveness and constant necessity for upgrading the knowledge base [27].

Most of the applied so far filtering approaches fall in two main categories: The *rule-based* method, which uses a set of heuristic rules to classify e-mail messages and the *statistical-based* approach which models the difference of messages statistically.

Text categorization techniques have become the dominant paradigm in building anti-spam filters due to their effectiveness and relatively low development cost [19]. Most of these research approaches attempt to classify mail on interesting and uninteresting ones, on basis of machine learning techniques [1] [15] [2] [10] [14] [3] [30]. Even though these techniques are characterized by high degrees of precision, they suffer from relatively lower accuracy ratings, meaning that they allow categorization of unsolicited mail as legitimate. Our approach proves superior to other machine-learning approaches in both means of accuracy and training times.

The rest of the paper is organized as follows: In section 2 we present a state-of the-art review in the area of email filtering. In section 3 our SF-HME system is introduced and presented in detail. In section 4 a discussion of the results that our method achieved is provided, as well as a comparative evaluation with other approaches. Section 5 concludes the paper.

2. RELATED WORK

Much focus has been attended recently in the area of email filtering and classification. Among other solutions, text-based filtering rises to prominence. In this section, we present a review and we attempt to classify research work on the area of spam filtering, according to the techniques applied. Sub-Section 2.1 presents systems, which filter emails by applying rule-based techniques. Sub-Section 2.2 is describing the statistical-based approaches, with major focus on Naïve-Bayesian, which has proved to be among the most effective in both means of accuracy and training costs [14] [19]. Sub-section 2.3 presents other approaches which belong to the area of artificial intelligence, such as artificial neural networks or genetic programming,which could

SAC'06, April, 23-27, 2006, Dijon, France.

not be classified in any of the previous categories. Sub-Section 2.4 presents works based on combinatory application of machine learning algorithms and their relative effectiveness comparison.

2.1 Rule-based approaches

Cohen [1] uses a system, which learns a set of keyword-potting rules based on the RIPPER rule-learning algorithm to classify emails into predefined categories. He reports a performance comparable to traditional TF-IDF weighting method. In general, building a rule-based system often involves acquisition and maintenance of a huge set of rules with an extremely higher cost compared to the purely statistical approach. Let alone such a system is hard to scale up.

Cunningham et al. [29] applied case-based reasoning, a method which has the advantage of being able to adjust in order to track concept drift, still though the reported experiments where on a very low number of test data, without many details of the characteristics of the used test data to be referenced. Additively, this method has the disadvantage of transferring the burden of labeling the data to the user.

Kolcz et al. [34] explored the impact of feature-based selection on signature-based classification. They explored by applying the I-Match algorithm the possibility of creating a server-side filter, by identifying spam messages through techniques of near-duplicate document detection, provided their hypothesis that spam often consists of highly similar messages sent in high volume. Still this technique is vulnerable to dedicated spamming attacks, such as frequent content alteration.

2.2 Statistical-based approaches

Statistical filters automatically learn and maintain these rules and easily adapt to new circumstances when new data arrive. The most popular and effective statistical spam filter is the naïve-Bayes spam filter.

Sahami et al. [2] analyzed a manually categorized mail corpus based on the use of words and phrases. In their research, they applied naïve Bayesian learning based on: words only, words and phrases, words-phrases and concurrent incorporation of domain specific characteristics, such as the inspection of the server domain of the sender (.edu, .gov etc.). They achieved high percentages of recall, especially for the latter case, which is based on characteristics that are added externally by the user and that cannot guarantee the accuracy of the results. For example the use of too many quotation marks might indicate spam but it might be dependent upon the specific authoring style of the sender.

Androutsopoulos et al. [14] [3] preprocessed manually categorized mail into four separate corpora using a lemmatizer and a stop list. Their investigation examines the effect of attribute-set size, training corpus size, lemmatization and a stop-list, that were not explored in Sahami et al.'s experiments [14]. Even though they achieved fairly high degrees of precision, their recall accuracy was rather low [30].

O' Brien et al. performed a comparative test of Naïve Bayes classifier versus Chi by degrees of freedom to classify spam mail, achieving an unimpressively lower recall [27].

Gee [30] applied latent semantic indexing analysis improving this, though this method was reported to suffer from serious errors, namely categorizing legitimate email as illegal, which consists to be an error with very high importance [14] [3] [10] [19].

Drucker et al. [10] analyzed their corpus by applying Ripper, Rocchio boosting and Support Vector Machines (SVM) and they found that SVM is somewhat lower in accuracy than boosting, but it dominates in the necessary training time.

Nicholas [22] applied a different boosting algorithm (Adaboost [31]) with decision stumps, trying to overcome the extremely slow training times of C4.5 that was examined by Drucker et al. [10], though the results didn't show up any superiority to the naïve Bayes method.

2.3 Other approaches

Drewes [32] created an artificial neural network based on email classifier; still the reported precision was significantly lower than that of other machine learning approaches. Furthermore, neural networks are not appropriate selection for this type of problem due to the extensive time they demand for training purposes [10].

Katirai et al., in [21], applied genetic programming algorithms, and additively performed a comparison of Naïve Bayes classifier. Even though the results on their set of emails were comparatively equal, there wasn't any obvious proof for a reason to substitute Bayesian filtering with genetic algorithms.

2.4 Algorithm efectiveness comparisons

Kiritchenko et al. [26] compared the performance of Naïve Bayes versus SVM, applying co-training on unlabeled data, and reported the superiority of SVM. Even though this method could be potentially preferable to users who are being released from labeling the data, still the reported accuracy is significantly less than the one recorded by other experiments [14] [3] [2].

Hidalgo [19] evaluated a number of algorithms, namely C4.5, Naïve Bayes, PART, Rocchio and SVM and did not distinguish any significant domination between the tested algorithms.

Carreras et al. [28] applied the AdaBoost algorithm [31] on a publicly available corpus - the PU1 corpus produced for the needs of the experiments described in [14] - and reported that this algorithm outperforms significantly the performance of Decision Trees and slightly the Naïve Bayes performance. Still as reported by the authors, the PU1 corpus is too small and too easy. Default parameters produced very good results and tuning parameters result only in slight improvements. For this reason we did not evaluate our results on the PU1 but on a much harder corpus, especially created for testing email filters.

Our approach is based on a combination of algorithms which have been applied effectively independently in the past for feature selection and classification purposes presenting high precision and accuracy ratings [5] [8]. For benchmarking purposes we have applied our method to a spam sample with very low discrimination potential between spam and non-spam samples, in order to prove the superiority of our method.

3. THE PROPOSED SF-HME SYSTEM

This paper presents a technique based on the Hierarchical Mixtures of Experts (HME) algorithm, which previously has been successfully applied on classification tasks [7] [8]. In order to

improve the classification accuracy of the algorithm, we applied on the training data a feature selection algorithm based on margin-selection strategy. Due to its application as a spam mail filter, we will refer to it as SF-HME system (Spam Filtering Hierarchical Mixtures of Experts) system. In the following paragraphs we describe the implementation choices.

3.1 Feature selection

Among the most challenging tasks in the classification process, we can distinguish the selection of suitable features to represent the instances of a particular class [4]. Additively, selection of the best candidate features can be a real disadvantage for the selection algorithm, in both means of effort and time consumption [6] [9].

We consider e-mails represented as vectors of binary features: $e=(f_1, f_2,.. f_N)$, where N is the number of features. For a given email, the feature f_j takes on value 1 if the email contains the feature and 0 otherwise.

We have decided to select features from all the available fields of an incoming email. Each term appearing in the body field is considered as a candidate feature. Moreover, terms from the other fields, like date of submission, address and name of the sender, subject, size and the X-Mailer field in the header of the html page, are used equiprobably as resources for selecting candidate features. In the context of supervised classification problems the relevance is determined by the given labels on the training data. A good choice of features is a key for building compact and accurate classifiers. From this very large number of candidate features the most relevant ones should be considered for efficient classification. This is consistent with many researchers [23] [24] [25], who found that systems using 1-3% of the total words in a category demonstrated little or no loss in performance.

The Iterative Search Margin Based Algorithm (Simba) has been applied in our case in order to select the most relevant features [5]. It operates based on a margin-based feature selection criterion to rank the features on the training set. The margins measure the classifier statistical confidence when making its decision. The Simba outperforms the other classical statistical approaches such as *relief* algorithm, *mutual information* criterion etc. [5]. There are two types of margins: *sample-margin* that measures the distance between the instance and the decision boundary induced by the classifier and the *hypothesis-margin* that requires the existence of a distance measure on the hypothesis class. The margin of a hypothesis with respect to an instance is the distance between the hypothesis and the closest hypothesis that assigns alternative label to the given instance.

The Simba algorithm finds the relevant features optimizing the *hypothesis-margin* for 1-Nearest Neighbor classifier. The result is a weighted vector: $w=(w_1, w_2 ,.. w_N)$, where N is the number of candidate features and each wj ranks the importance of feature ff in the classification task.

For a training set of instances P, in our case e-mails, it is easy to calculate the hypothesis margin for an instance $x \in P$ using the following formula:

$$\theta_P(x) = \frac{1}{2}(\| x - nearmiss(x) \| - \| x - nearxit(x) \|) \quad (1)$$

where *nearhit(x)* and *nearmiss(x)* denote the nearest point to x in P with the same and different label, respectively. Note that a chosen set of features affects the margin through the distance measure.

The algorithm at the start point initializes the weighted vector $w = (1, 1, ...,1)$ and in a number of iterations T, using a stochastic gradient ascent over the sum of $\sum\theta_p(x_i)$ for all the instances x_i, it updates the vector w: $w=w+\Delta$, where vector Δ is calculated from the following equation:

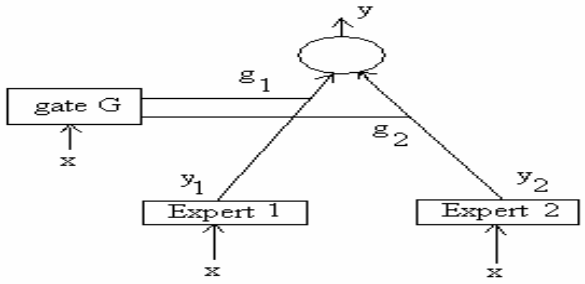
$$\Delta i = \sum_{x \in P} \frac{\partial \theta(x_i)}{\partial w_i} = \frac{1}{2}\sum_{x \in P}\left(\frac{(x_i - nearmiss(x_i))^2}{\| x - nearmiss(x) \|_w} - \frac{(x_i - nearhit(x_i))^2}{\| x - nearhit(x) \|_w}\right) \quad (2)$$

The algorithm finally converges in a typical number of iterations resulting in a weighted vector w containing the relevancy ranks for the features.

3.2 Hierarchical Mixtures of experts Algorithm

A modular approach of neural networks known as *Mixture of Experts* (ME) has attracted quite attention for solving problems in machine learning. The hierarchical ME models have been successfully applied to classification problems [8] [13].

MEs try to solve the problems using a divide-and-conquer strategy by decomposing the whole, usually complex problem into simpler sub problems. MEs belong to the class of probabilistic models [9] and consist of a set of *experts*, which model conditional probabilistic processes, and a *gate*, which combines the probabilities of the experts. The gating network of ME learns to classify the input space into patterns, in a soft way, so permitting overlaps and attributes expert networks to these different patterns. Figure 1 shows a mixture of expert's model of two experts and one gate. The standard choices for experts are generalized linear models [7] and multilayer perceptrons [11]. Here we use the generalized linear models the mathematical form of which is $y_i = w_i^T x$, where w_i parameters. The output for the above network is the weighted (by the gating network outputs) mean of the expert outputs.



$y(x) = \sum_i g_i(x) y_i(x)$ (2), Where $g_i(x)$ denotes the probability that input x is attributed in expert i. In a classification problem we are always interested to compute the a-posteriori probability of class label y given the evidence x. Otherwise, in terms of a ME model the conditional probability $p(y|x)$ of the

output y given the input x. This can be formulated by equation (3): $p(y \mid x) = \sum_i g_i(x)\phi_i(y \mid x)$ (3), where φ_i (y_i in the shape) represent the conditional densities of target *Y* given the expert *i*. In order to ensure a probabilistic interpretation to the model, the activation function g_i of the gate is chosen to be the soft-max function [12]:

$g_i = \exp(z_i) / \sum_j \exp(z_j)$ (4), where z_i are the gating network outputs before thresholding. By this function, the gating network outputs sum to unity and are non-negative.

The ability to model non-linear functions is a desirable one in statistical models. However, the non-linear functions that a ME model can represent are somewhat restricted since the gate can only form linear boundaries between adjacent expert regions in the input space. A complementary approach proposed by Jordan and Jacobs [7] is to use experts, which are *themselves* mixtures-of-experts models. This approach is easily implemented as a generalization of the mixture of experts model. The result is known as *hierarchical* mixtures-of-experts model (HME) and may be visualized as a tree structure. Such a model is shown in

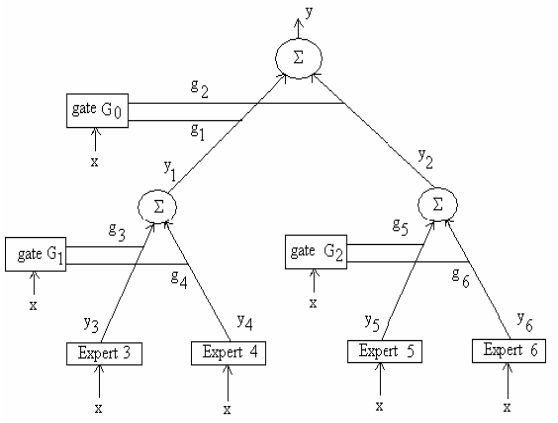


Figure 2. Tree structure of a hierarchical mixture of experts with binary branches at each non-terminal node, and a depth of 2

Figure (2). The architecture of these models consists of two levels of gates with binary branches at each non-terminal node. The output of the terminal experts E_3, E_4, E_5, E_6 are y_3, y_4, y_5, y_6 respectively, the outputs of the gates G_1, G_2 rooted at the non-terminal nodes in the second level are g_3, g_4, g_5, g_6. For the outputs of the non-terminal nodes in the second we have $y_1=g_3y_3+g_4y_4$, $y_2=g_5y_5+g_6y_6$ and the finally, the output of the system is $y=g_1y_1+g_2y_2$;

The training phase, which aims in estimation of system parameters, is considered of vital importance for a classification

Figure 1. A mixture of experts model consisting of two experts E_1, E_2 and one gate G

system. For the purposes of our classification task , the model must be trained over a suitable number of training instances in order to estimate the parameters, i.e. the functions g_i, φ_i. As aforementioned, for g_i we use the soft-man function (equation 4) and for experts generalized linear models. The distribution of equation (3) forms the basis for the mixture of experts' error function, which can be optimized using gradient descent or the Expectation -Maximization (EM) algorithm [7], but here we use the EM algorithm.

The EM algorithm functions in an iterative way in problems where data is missing or hidden. In the case of mixture of expert's models, missing data is considered the outputs of experts. Moreover, EM is an attractive method for training since it enables the optimization of a ME or HME model to be broken up into a set of optimizations, one for each expert and gate. It is commonly used to train Gaussian mixtures and other mixture models. The principle of maximum likelihood is a standard way to motivate error functions. Given a set of independently distributed training data $\{x^n, t^n\}$, $n=1..N$, the likelihood L of the data is given by: $L = \prod_n p(x,t) = \prod_n p(t \mid x)p(x)$. Taking the negative logarithm of the likelihood and dropping the term $p(x)$ (because it does not depend on the model parameters) we can obtain a cost function $E = -\sum_n p(t \mid x)$. Taking into account equation (3), the cost function for this classification task can be formulated as follows: $E = -\sum_n \ln \sum_i g_i(x)\phi_t(t \mid x)$ (5). This cost function must be minimized to find the optimal parameters using the EM algorithm, a complete description of which can be found in [7].

4. SF-HME SYSTEM EVALUATION

Our experiments were based on a publicly available corpus, provided by the Open Project SpamAssasin for evaluation purposes and benchmarking of unsolicited bulk email filters [35]. In recent bibliography very few databases have been publicly available for evaluation purposes. For some of them the reader may refer to [19] [36]. One of the most extensively exploited corpora is the PU1 email corpus [28], collected for the experiments described in [3] [14]. We have included several characteristics for classification purposes in our experiments - that have all been removed from the PU1 corpus - such as the presence of HTML code, which makes hard to discriminate spam from legitimate messages. Furthermore, in order to handle the privacy issues rising when it comes to mail corpora, the PU1 corpus has been encrypted prior to publicizing and therefore has reduced processing capabilities; for example it is not appropriate for co-processing with lexical thesauri or ontological processing etc. In order to overcome these limitations, the samples we used are not encrypted, and can be freely downloaded from [35].

4.1 Evaluation on a publicly available corpus

4.1.1 Sample data characteristics

Our experiments were held by applying our SF-HME method to a large public spam corpus, described in the previous paragraph. This is a selection of mail messages, created especially for benchmarking of spam-filtering systems. The most recent collection *20030228_spam_2* has been selected for our experiments. The legitimate corpus consists of two collections:

the *20030228_hard_ham_*2 and *20030228_easy_ham* containing 250 and 2500 non-spam messages respectively.

The *hard_ham_2* corpus contains non-spam messages that are difficult to be discriminated from spam messages because of their high similarity to typical spam, obvious by the presence of several features: use of HTML, unusual HTML markup, colored text, "spammish-sounding" phrases etc. The *easy_ham* corpus contains non-spam messages that are typically very easy to be discriminated from spam messages, since they do not contain any spammish signatures (like html etc).

4.1.2 Experimental details

In order to test the robustness of our SF-HME system (especially the Simba feature selection strategy coupled with the HME classification algorithm), we scanned html code from these corpora and extracted everything that can be used for a candidate feature for discrimination (fields like *received_from*, *delivery date*, *message-id*, X-*keywords*, *Content Type*, *subject*, *body*, *size* and many other type of information like html tags for *fonts* and *colors*, URL's for multimedia resources and features from *java scripts* code etc). We believe all those features are extremely useful in the discrimination procedure, so we included them in the feature selection stage. Avoiding allowance of any simplification for the discrimination procedure, we did not mix the two non-spam corpora to make a single non-spam corpus, but we performed two separated experiments one for each corpus.

For the *easy_ham* corpus our algorithm performed as it was expected extremely excellent achieving 100% discrimination accuracy. We describe below the followed process when experimenting with the *hard_ham* corpus.

We divided the 1397 spam messages of the *20030228_spam_2* collection into 5 groups, each group containing 240 messages (150 for training and 90 for testing). From the 250 messages of the *hard_ham_2* corpus the 150 were used for training and the 90 for testing (we used 90 because our program separated only 243 discrete emails from the hard_ham_2 corpus).

Combining each *spam* group with the *hard_ham* corpus, we performed 5 evaluation experiments using as evaluation measures the average *precision* and *recall*.

All measures given in Table 1 are averaged into five groups.

Total features: 515,219. Discrete features: 31,628.

We selected the 300 most representative features by the Simba feature selection algorithm after stemming - a technique that has been proved to enhance email-filtering efforts [14] - and conversion to lower case and removal of punctuation marks.

The loglikelihood before learning was: -182.028879. The loglikelihood after 3 only iterations of the EM algorithm - 0.000957. Table 2 summarizes the results from the first experiment.

Table 1. The 20 most representative features for the classification task as selected by Simba feature selecting algorithm

Feature	Simba score	Feature	Simba score
netnoteinc	1	Deliveri	0.4086
2002	0.61775	http	0.34849
yyyy@netnoteinc	0.60678	Uid	0.34538
taint	0.561	Copyright	0.29373
postfix	0.55541	0000	0.2922
2001	0.5481	Keyword	0.28825
Tm	0.5096	v1	0.26284
text/plain	0.44924	Subscript	0.25903
newslett	0.43718	Juli	0.25345
//www	0.4086	Qmail	0.24557

Table 2. Recall and Precision ratings achieved in our experiments for legitimate and spam mail

	Recall	Precision
Spam	92.22%	80.58%
Legitimate	77.78%	90.91%

4.1.3 Results and discussion

As the recorded results show, there are strong indications about the robustness of the applied method through our experiments (legitimate emails are very hard to discriminate from spam in the corpus we used). Other research attempts present very high precision [14] [3] [19], but on test data with low similarity between legitimate and spam mail, which makes the classification process easy task with little or no effect when applying tuning parameters [28]. Additively, on most recent versions of the same corpus, by applying SVM, lower degrees of precision and recall have been reported [33]. Still, on this updated version of the corpus, HTML comments and formatting tags have been removed which is not the case for the hard_ham corpus that has been used for our evaluation purposes.

Our system presents high degrees of precision, considerably higher than rule based or even relative to Bayesian-based filtering and additively has the advantage that it demands a small time of training on a small amount of corpora. The number of representative features can be updated periodically and kept separate from other data. We intend to expand our experiments with different combinations of algorithms in the future.

5. CONCLUSIONS

Based on performing experiments with publicly available data-sets, with high similarity between legitimate and unsolicited mail, we came up to the following conclusions: Our SF-HME approach proves to be robust and efficient in both means of accuracy and training time. Furthermore, it does not suffer from the necessity of reconstructing the training set, as it happens with other approaches [31]. In our experiments we examined more characteristics than that of other approaches, which removed attachments, HTML tags and other characteristics, simplifying the discrimination process [14]. We achieved results that outperform the Naïve Bayesian classifier which has been in general approved as one of the most efficient ones [14] [30] [28] [19]. We are planning to experiment in the future with a broader combination of algorithms and to experiment with our techniques in

identification of emails from same author among a collection of emails (for forensic reasons).

6. ACKNOWLEDGMENTS

This work was co-funded from the E.U. by 75% and from the Greek Government by 25% under the framework of the Education and Initial Vocational Training Program – Archimedes.

7. REFERENCES

[1] Cohen, W. W. Learning Rules that Classify E-mail. In Proceedings of the 1996 AAAI Spring Symposium on Machine Learning in Information Access, California.

[2] M. Sahami, S. Dumais, D. Heckerman, and E. Horvitz. A Bayesian Approach to Filtering Junk E-Mail. Learning for Text Categorizaton - Papers from the AAAI Workshop,1998, pages 55-62, Madison Wisconsin. AAAI Technical Report WS-98-05.

[3] I. Androutsopoulos, J. Koutsias, K. V. Chandrinos, G. Paliouras, and C. D. Spyropoulos. An Evaluation of Naïve Bayesian Anti-Spam Filtering. In Proc. of the workshop on Machine Learning in the New Information Age, 2000.

[4] Kira, K., & Rendell, L. (1992). A practical approach to feature selection. In Proc. 9th International workshop on machine learning (pp. 249-256)

[5] Gilad-Bachrach, Navot A., Tishby N. Margin Based Feature Selection - Theory and Algorithms. In Proc of ICML 2004

[6] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. Neural Computation, 3(1):79--87, 1991.

[7] Michael I. Jordan and Robert A. Jacobs. Hierarchical mixtures of experts and the EM algorithm. Neural Computation, 6(2):181--214, 1994.

[8] S. R. Waterhouse and A. J. Robinson. Classification using hierarchical mixtures of experts. In Proceedings 1994 IEEE Workshop on Neural Networks for Signal Processing, pages 177--186, Long Beach CA, 1994. IEEE Press.

[9] Jacobs, R. A., Jordan, M. I., Nowlan, S. J. and Hinton, G. E. 'Adaptive mixtures of local experts', *Neural Computation* **3**(1), 79–87, 1991.

[10] H. Drucker, V. Vapnik, mad D. Wu. Support Vector Machines for Spam Categorization. *IEEE Trans. on Neural Networks,* 10(5), 1999.

[11] Andreas S. Weigend, Morgan Mangeas, and Ashok N. Srivastava. Nonlinear gated experts for time series: Discovering regimes and avoiding overfitting. International Journal of Neural Systems, 6:373--399, 1995.

[12] J. S. Bridle. Probabilistic interpretation of feed forward classification network outputs with relationships to statistical pattern recognition. In F. Fogelman Souli'e and J. H'erault, editors, Neurocomputing: Algorithms, Architectures, and Applications, pages 227--236. Springer Verlag, New York, 1990.

[13] Jurgen Fritsch, Michael Finke, and Alex Waibel. Context-dependent hybrid HME/HMM speech recognition using polyphone clustering decision trees. In Proceedings of ICASSP-97, 1997.

[14] I. Androutsopoulos, J. Koutsias, K. Chandrinos, and C. Spyropoulos. An experimental comparison of naïve Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In Proc. *of SIGIR,* 2000.

[15] Jake D. Brutlag and Christopher Meek. Challenges of the Email Domain for Text Classification. In *Proc. of the 17th International Conference on Machine Learning*, pages 103–110, Stanford University, USA, 2000.

[16] Cranor, L. and Lamachia, B. (1998), Spam! *Comm. ACM 41*, 8, 74–83.

[17] Gburzinsky, P. and Maitan, J. (2004) Fighting the Spam Wars: A Remailer Approach with Restrictive Aliasing, ACM Transactions on Internet Technology, vol. 4, No 1, Feb. 2004, pg. 1-30.

[18] Stephen Hinde: Spam, scams, chains, hoaxes and other junk mail. Computers & Security 21(7): 592-606 (2002).

[19] Hidalgo J. (2002), Evaluating Cost Sensitive Bulk Email Categorization, pp 615-620, SAC 2002, Madrid, Spain.

[20] P. Hoffman and D- Crocker. Unsolicited bulk email: Mechanisms for control. Technical Report UBE-SOL, IMCR-008, Internet Mail Cons., 1998.

[21] H. Katirai. Filtering junk e-mail: A performance comparison between genetic programming & naïve Bayes. Available: http://members.rogers.com/hoomank/papers/katirai99filtering.pdf , 1999.

[22] Nicholas T., 2003. "Using AdaBoost and Decision Stumps to Identify Spam E-mail", Available: http://nlp.stanford.edu/courses/cs224n/2003/fp/tyronen/report.pdf

[23] Lewis, D. D, Feature selection and feature extraction for text categorization, Morgan Kaufmann, San Francisco, pp. 212-217, 1992.

[24] Koller, D. and Sahami, M., Hierarchically classifying documents using very few words, in I*nternational Conference on Machine Learning (ICML)*, pp. 170-178, 1997.

[25] Mladenic, D, Feature subset selection in text-learning, in *Proc. of the 10th European Conference on Machine Learning*, 1998.

[26] S. Kiritchenko and S. Matwin, "Email Classification with Co-Training," in Proc. Annual IBM Centers for Advanced Studies Conference (CASCON 2001).

[27] O'Brien and Carl Vogel Spam Filters: Bayes vs. Chi-squared; Letters vs. Words. Presented at the International Symposium on Information and Communication Technologies, September 24-26, 2003

[28] X. Carreras and L. Marquez. Boosting trees for anti-spam email filtering. In Proceedings of RANLP-01 International Conference on Recent Advances in Natural Language Processing, Tzigov Chark, BG, 2001

[29] Cunningham P., Nowlan N., Delany S. J., Haahr J. " A Case-Based Approach to Spam Filtering that Can Track Concept Drift" In The ICCBR'03 Workshop on Long-Lived CBR Systems, Trondheim, Norway, June 2003

[30] Gee K. Using Latent Semantic Indexing to Filter spam, SAC 2003, Florida, USA

[31] Shapire R. E., Singer Y. "Improved boosting algorithms using confidence-rated predictions. Machine learning 37(3): pp. 297-336, 1999

[32] Drewes R. An artificial neural network spam classifier, available at project homepage: www.interstice.com/drewes/cs676/spam-nn

[33] Woitaszek M., Shaaban M., "Identifying Junk Electronic Mail in Microsoft Outlook with a support vector machine", proc. of the 2003 symposium on applications and Internet

[34] Kolsz A., Chowdhury A., Alspector J. "The impact of feature selection on signature-driven spam detection". Conference on email and Anti-Spam 2004, CA, USA

[35] http://spamassassin.org/publiccorpus

[36] Fawcett T. "In vivo" spam filtering: A challenge for KDD. SIGKDD explorations, vol 5,issue 2, 2003, pp. 140-149.