Al-Khwarizmi Engineering Journal, Vol. 6, No. 2, PP 83-92 (2010)

Al-Khwarizmi Engineering Journal

# Software Engineering-Based Design for a Bayesian Spam Filter

**Mumtaz Mohammed Ali AL-Mukhtar**
*College of Information Engineering/AL-Nahrain University*
Email: mumtaz_almukhtar@yahoo.com

(Received 17 March 2008; accepted 9 September 2009)

## Abstract

The rapid spread and the easy availability of a free e-mail service have made it the medium of choice for the sending of unsolicited advertising and bulk e-mail in general. These messages, known as junk e-mail or spam mail, are an increasing problem to both Internet users and Internet service providers (ISPs).

The research resolves one aspect of the spam problem by developing an appropriate filter for the e-mail client. The proposed filter is a combination of three forms of filters: Whitelist, Blacklist, and a Bayesian filter. Whitelist-based filter only accepts e-mails from known addresses. Blacklist filter blocks e-mails from addresses known to send out spam. Bayesian content-based filter makes estimations of spam probability based on the text and filters messages based on a pre-selected threshold.

The Bayesian filter is selected to be the main filter. The Bayesian filter is manually trained on a set of gathered e-mails; some of them are spam and the others are legitimate based on the contents of an e-mail. Thereafter the classification phase has been implemented for new entered e-mails. All the required databases are constructed in form of tables stored in the Structured Query Language (SQL) server. The filter at the client side can transparently access the database in order to carry on the intended filtering. The proposed system (e-mail client interface and the filters) can manage messages written in both Arabic and English languages which is crucial for the users in our region.

Software engineering principals are implemented throughout the design process to make the system less vulnerable to faults and easily maintained. The design steps have followed the Waterfall-model using the ASCENT software. A user-friendly interface has been developed to access the features of the spam filter at the client side. Visual Basic version 6 has been used to develop the system. As well, the SQL server has been implemented to build and process the database.

A number of performance measurements have been carried out with asset of gathered e-mails. The results are used to evaluate the performance of the filter and to prove its efficiency.

***Keywords***: *Spam, client e-mail, bayesian filter, SQL server, waterfall model.*

## 1. Introduction

In the past few years, Internet Technology has affected our daily communication style in a radical way: the electronic mail (e-mail) concept is used very extensively for communications nowadays. This technology makes it possible to communicate with many people simultaneously in a way so easy and cheap that it is currently considered the first worldwide into business sector [1].

However, the abuse of e-mails has the drawback that the volume of e-mails that show up in mailboxes has been exponentially increasing. Moreover, many e-mails are received by users without there desire: “spam mail” (or “junk mail” or “bulk mail”) is the general name used to denote these types of e-mail. Spam mails, by definition, are the electronic messages posted blindly to thousands of recipients, usually for advertisement, and represent one of the most serious and urgent information overload problems [1, 2].

Spam has caused some serious problems. Firstly, it wastes a mass of network resources that are very important for network users, especially those in enterprises or corporations. People need to spend a lot of time to deal with spam every day. Even worse, many current spam mails bring users

unexpected malicious attachments which would seriously crack the user's system. Therefore, spam is a headachy problem [3].

Spam filtering (i.e., distinguishing between spam and legitimate e-mail messages) is a commonly accepted technique for dealing with spam [4]. Spam filters vary in functionality from black-lists of frequent spammers to content-based filters. The latter are generally more powerful, as spammers often use fake addresses. Existing content-based filters search for particular keyword patterns in the messages. These patterns need to be crafted by hand, and to achieve better results they need to be tuned to each user and to constantly maintained [5,6].

In this paper the spam filtering has been addressed with the aid of Bayesian classifier, which learn to identify spam e-mail after receiving training on messages that have been classified as spam or non-spam (legitimate).

## 2. E-mail System Components

E-mail is a set of mechanisms designed to allow computer users send messages to one another. At the most basic level the e-mail system can be divided into two principles components [7]:

A. E-mail Servers: which are hosts that deliver, route, and store e-mail messages.
B. E-mail Clients: which interface with users and allow users to read, compose, send, and store e-mail messages.

At the lowest levels, it can get quite complicated. At the highest level it is quite simple, consisting of mail messages, a protocol for moving those messages from place to place, and an interfaces for users to perform various related tasks.

An e-mail message by itself is of a limited usefulness. There is a need to be a way to move it from one location to another. This work is divided into several tasks [8]:

A. MTA (Mail Transfer Agent): that routes e-mail.
B. MDA (Mail Delivery Agent): that delivers it on behalf of an MTA.
C. MUA (Mail User Agent): which provides an interface for the user to send and receive messages.

## 3. E-mail protocols

To carrry out the project two main e-mail protocols are required which are SMTP, POP3 [9].

### A ) SMTP (Simple Mail Transfer Protocol)

It is implemented as a communication protocol between two machines, a client and a server. SMTP is also a store and forward protocol, meaning that it permits a message to be sent through a series of servers in order to be delivered on an end destination. Servers store incoming messages in a queue to wait attempts to transmit them to the next destination. The next destination could be a local user or another mail server.

### B ) POP3 (Post Office Protocol)

This protocol allows a client machine to connect to a remote mail server, and download any mail to a local mailbox. At the highest level, POP3, like SMTP, is implemented as a communication protocol between two machines: a client and a server [9].

## 4. Proposed Spam filter

The proposed filter is designed to be a combination of three layers that work altogether. White-list layer only accepts e-mails from known addresses. Black-list layer blocks e-mails from addresses known to be spam. Content-based filter makes estimation of spam likelihood based on the text of the e-mail and filter messages according to a pre-selected threshold. The content based filter uses a Bayesian algorithm to estimate the probability of a message being a spam.

A database is constructed in form of tables stored in SQL server. The filter at the client side can transparently access the database in order to carry on the intended filtering.

### 4.1 Whitelist versus Blacklist

The whitelist makes a decision according to the message sender address. It compares this message address to the table in the SQL server that contains all of the message addresses that are to be accepted by the user.

However, the blacklist compares the message address to the table in the SQL server that

contains all message addresses that are rejected by the user. When the user receives a spam message, he can add the address of the sender to his blacklist.

### 4.2 Bayesian Algorithm

1) Split e-mail in tokens.
   i. Need number of messages for spam and legitimate.
   ii. Need frequency of each word for each type.
2) Calculate probabilities
   i. P(legitimate) = word frequency /number of legitimate messages.
   ii. P(spam) = word frequency/ number of spam messages.
3) Calculate likelihood of being spam (spamicity) using a special form of Bayes' Rule where likelihood = a/(a + b), where a is the probability of a legitimate word and b is the probability of spam word.
4) Choose tokens whose combine probability is farthest from 0.5 either way. This is because the farther it is from 0.5 (neutral), with more certainty we can say it belongs to either strategy.
   i. Do this for n numbers for instance choose to have 15 extremes.
   ii. Combine their probability to get a figure for message using Bayes' Rule. In basic terms, Baye's Rule determine the probability of an event occurring based on the probabilities of two or more independent evidentiary events. For three evidentiary events a, b, and c, the probability is equal to [10]:

$$\frac{a\ b\ c}{a\ b\ c + (1-a)*(1-b)*(1-c)} \quad \ldots(1)$$

   In this fashion, the rule can be expanded to accommodate any number of evidentiary events.
   iii. If the end result is closer to 1.0, then the message is classified as spam, and if it is closer to 0.0, the message is classified as legitimate. The cutoff range that has been specified for spam is that it should be greater than 0.85, but spam results are above 0.98.

## 5. Spam Blocking System Design

Data flow diagrams are developed to give the details of constructing the whole spam blocking system. These data flow diagrams are constructed according to the waterfall model using the ASCENT software [11]. The system design is divided into 6 levels, which describe the operations (processes) involved and how data flow across the system. Figure 1 depicts the basic system level (level 0).

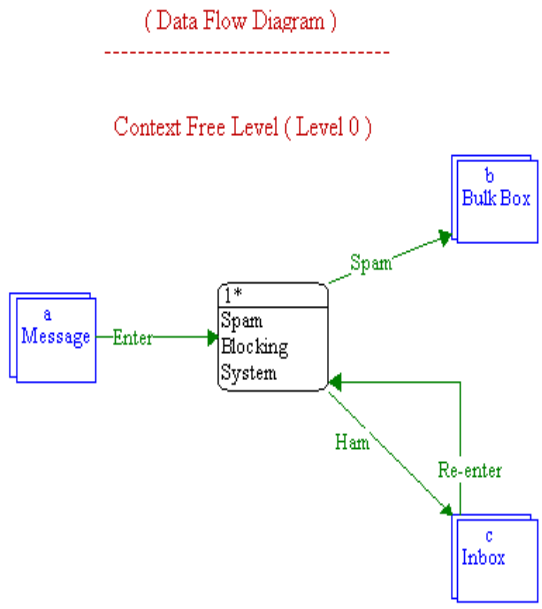


**Fig.1. System level 0.**

When a message enters the system, a decision has to be made by the spam blocking system where to put the message. If the message is spam, it will be submitted to the bulk box, otherwise (i.e., legitimate message) it is submitted to the inbox. Re-enter direction means that a spam message has been wrongly treated as legitimate, which causes the spam message to enter the inbox as the blocking system could not discern it correctly. However, it is up to the user to set this message as a spam and add it to the training corpus. So, for the next time if this message or a similar one enters the system, it will be recognized and added to the bulk box. A star symbol has been used in figure 1 and next figures to indicate that the designated block comprises more details to be exploded in the next level.

Figure 2 illustrates level one in the system data flow diagram. This level has three processes:

- Mail Server: accepts the message from the client and delivers it to another client according to the e-mail address.
- Client: receives the message from the mail server and decides whether it is a spam to send to the bulk box or it is a legitimate to send to the inbox.

- SQL Server: this process contains the database, which the client relies on to make the decision concerning the received message. So, by this process the database is read, updated, and changed.

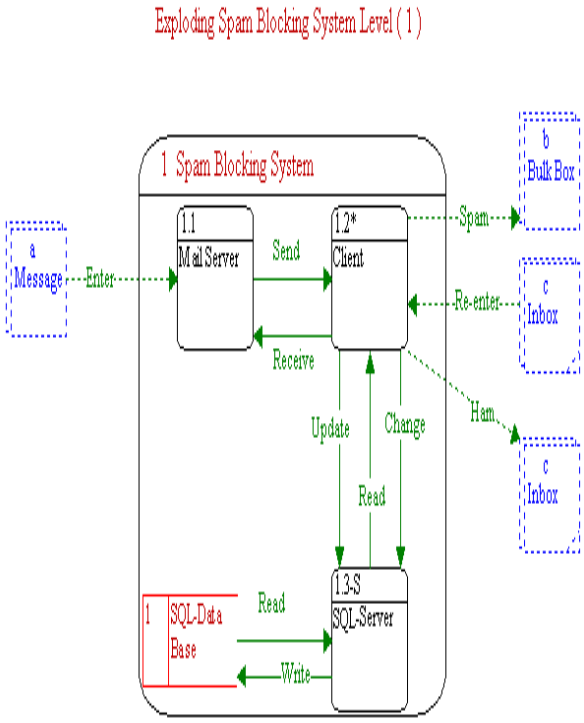


**Fig.2. Level 1.**

Level 2 which is depicted in figure 3 is an exploding of the client process that contains two processes:

- Mail User Agent (MUA): sends and receives messages from the mail server, and it is responsible for providing an interface for users to manage their mail. This management typically includes viewing messages, managing mail folders, and composing new e-mails, as well as replying to a message and sending an existing message to other users.
- Filtering Program: checks the received message from the MUA whether it is a spam or ham. Thereafter the SQL server database is updated.

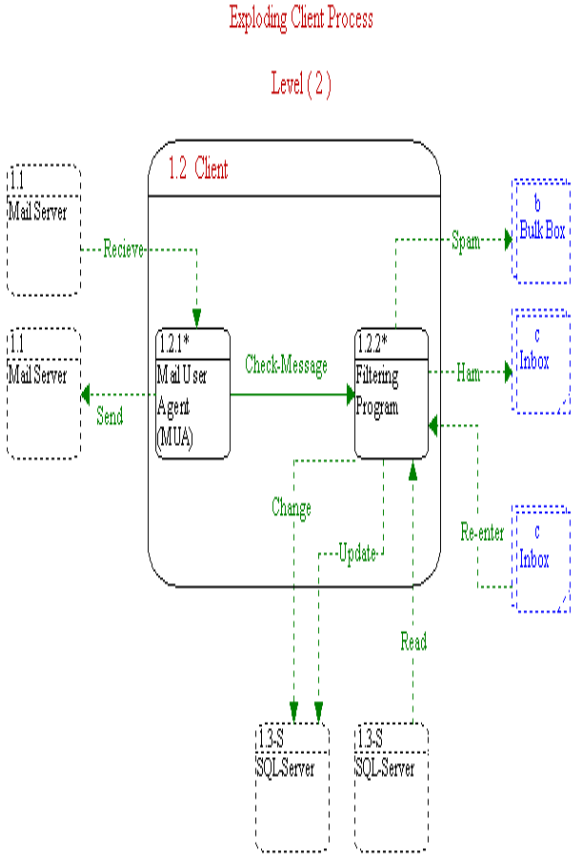


**Fig.3. Level 2.**

Level three, which is exploding of the MUA is illustrated in figure 4. It is comprised of two processes:

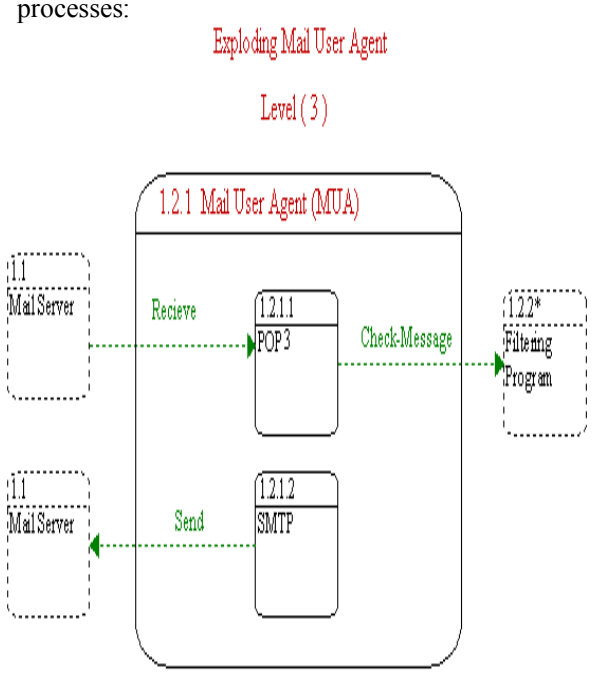


**Fig.4. Level 3.**

- POP3: provides a protocol for MUA to download the user's mailbox from a remote server.
- SMTP: implements the communication protocol between two machines that is a client and a server. In its simplest form, a message is sent from a user on one machine to a mailbox in another machine.

Figure 5 depicts level four, which is the exploding of white list and blacklist checking that contains two processes:

- Whitelist Checking: in this process the message ID is compared with the one stored in the SQL server, which contains the addresses of the IDs of allowed users. If a match occurs, the SQL server database will be updated and the message is added to the inbox. Otherwise, in case of no match, the ID will go through another checking which is the black list.
- Blacklist Checking: this process makes a comparison between the messages ID that is not found in the white list table in the SQL server database with the one that contains all the addresses that are not accepted by the user. If a match occurs, the SQL server database will be updated and the message is added to the bulk box. Otherwise, the message enters to the reduction message process.

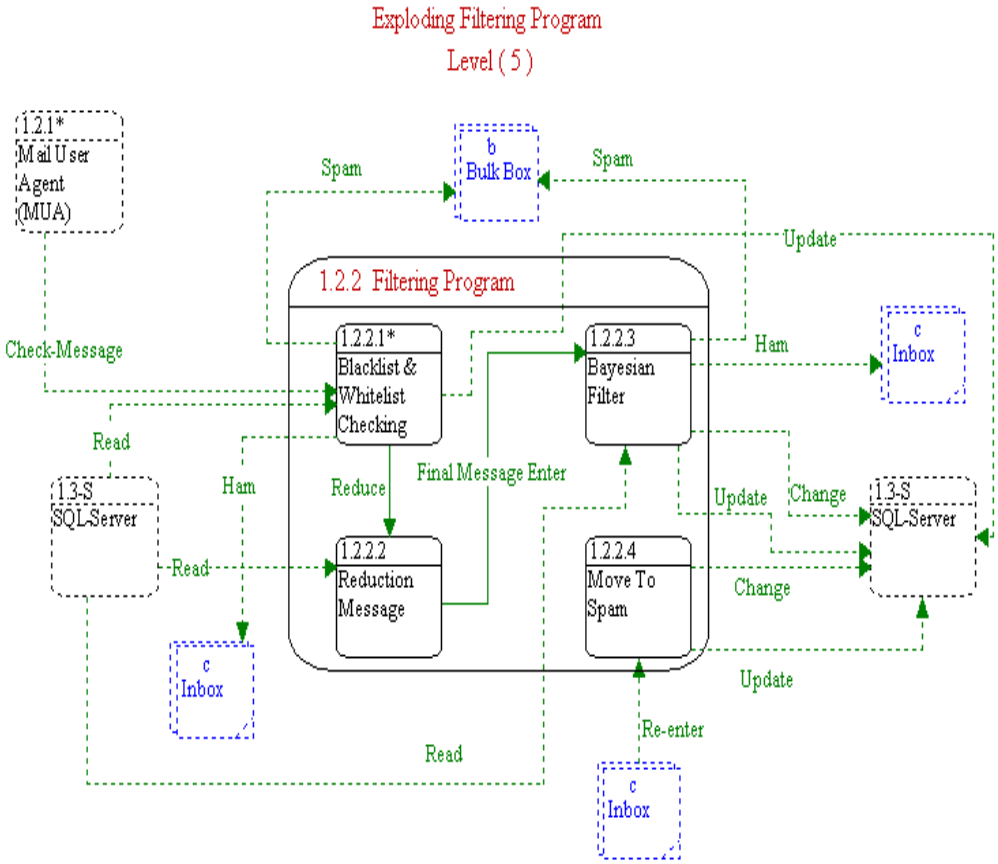


**Fig.5. Level 4.**

Figure 6, which constitutes level five, is the exploding of filtering program that contains four processes, which are:

- Black List & white List Checking: this process is mentioned earlier in level four.
- Bayesian Filter: this process is the main part in the blocking system.
- It does several activities: accepts the message from the reduction process, reads the database from the SQL server to make its decision for spam or legitimate, updates the database for the existence words in the message, and changes the database if a false positive
- Move to Spam: sometimes the message enters the inbox as legitimate, but it is really misclassified by the filtering, i.e., a false positive case has encountered.

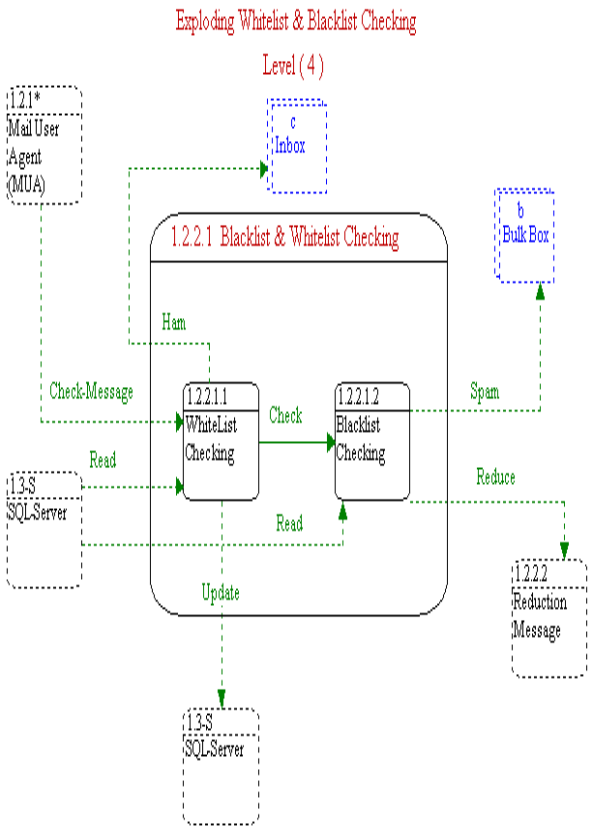


**Fig.6. Level 5.**

In this case the user could decide that such a message will be considered as a spam in future classifications by activating the re-enter the message to the filter again. So, the database in the SQL server will be updated accordingly.

## 6. Performance Measures

The performance measures used in this paper are introduced. Let $S$ and $L$ stand for spam and legitimate message, respectively. ${}^{n}L\rightarrow L$, ${}^{n}S\rightarrow S$ denote the numbers of legitimate and spam messages correctly classified by the system i.e., true positive (TP) and true negative (TN) respectively. ${}^{n}L\rightarrow S$ represents the number of legitimate messages misclassified as spam i.e., false positive (FP), and ${}^{n}S\rightarrow L$ is the number of spam messages wrongly treated as legitimate i.e., false negative (FN). Then spam precision (P), and spam recall (R) are defined as follows [12]:

$$P = \frac{TP}{TP + FP} \qquad ...(2)$$

$$R = \frac{TP}{TP + FN} \qquad ...(3)$$

When filtering spam, it is worth noting that misclassifying a legitimate mail as spam is much more severe than letting a spam message pass the filter. Letting a spam go through the filter generally does no harm while blocking an important personal mail as spam can be a real disaster. The usual precision/recall measures tell little about a filter's performance when false positive and false negative are weighted differently. To introduce some cost sensitive evaluation measures that assign false positive a higher cost than false negative, a weighted accuracy (WAcc) and weighted error rate (Werr =1-Wacc ) measures specially tailored for this scenario can be used. Wacc *and* Werr was introduced by Androutsopoulos et al. [13] as:

$$WAcc = \frac{\lambda \cdot n_{L\rightarrow L} + n_{S\rightarrow S}}{\lambda \cdot NL + NS} \qquad ...(4)$$

$$WErr = \frac{\lambda \cdot n_{L\rightarrow S} + n_{S\rightarrow L}}{\lambda \cdot NL + NS} \qquad ...(5)$$

where $NL$ is the total number of legitimate messages, and $NS$ denotes the total number of spams. WAcc treats each legitimate message as if it were $\lambda$ messages: when false positive occurs, it is counted as $\lambda$ errors; and when it is classified correctly, this counts as $\lambda$ successes. The higher $\lambda$ is, the more cost is penalized on false positives. Androutsopoulos et al. [11] also introduced three different values of $\lambda$: $\lambda$ = 1, 9, and 999. When $\lambda$ is set to 1, spam and legitimate mails are weighted equally; when $\lambda$ is set to 9, a false positive is penalized nine times more than a false negative; for the setting of $\lambda$ = 999, more penalties are put on false positive: misclassification a legitimate mail is as bad as letting 999 spam messages pass the filter. Such a high value of $\lambda$ is suitable for scenarios where messages marked as spam are deleted directly.

In practice, when $\lambda$ is assigned a high value (such as $\lambda$ = 999), WAcc can be so high that it tends to be easily misinterpreted. To avoid this problem, it is better to compare the weighted accuracy and error rate to a simplistic baseline i.e., legitimate messages are never blocked and spams can always pass the filter.

## 7. System Evaluation

To examine the performance of the proposed spam blocking system, a testing corpus of 800 e-mails; 400 for spam and 400 for ham e-mails has been used. In order to evaluate the effect of Whitelist, Blacklist, and the Bayesian filter that are used for detecting spam, each layer is tested separately as shown in figure (7).

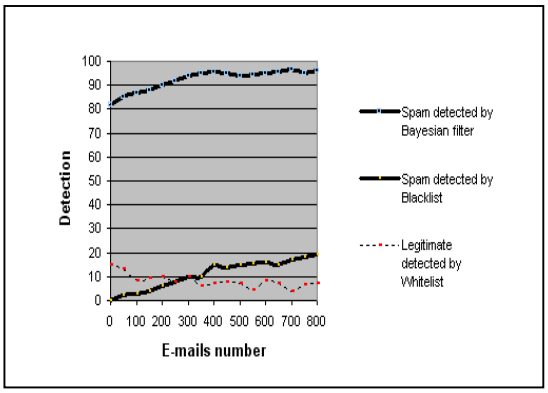


**Fig.7. Spam Filter Detection Versus Layer.**

The results obtained are summarized as follows:

- **Spam Detected by Bayesian Filter:** the system is in a state of testing and this means that the system has already been trained. For that reason the detection percentage starts from a good percentage, which is $\cong$ 82%. Thereafter, as more learning is carried out, spam detection could approach 100% percentages.
- **Spam detected by Blacklist:** this method depends on the e-mail addresses of spammers that are stored in the Blacklist table. Initially it starts from zero and as much as the new e-mails are entered as much as the detection increases. That is because when the new e-mail is detected as spam its address is automatically saved in a Blacklist table.
- **Legitimate detected by Whitelist**: this method depends on the e-mails headers that are addresses of non-spammers. The detection line starts from $\cong$ 10% and it vibrates around this percentage. This is because of that the new e-mails entered are from persons that were initially entered in the whitelist.

It may be concluded that the Bayesian filter had detected more Spam percentages rather than the other two methods. This is because the Bayesian filter tests the message body rather than the e-mail addresses.

Figure (8) reveals that the spam detection percentage increases with the increasing number of incoming e-mails. This is because the statistical filter would learn more with more various features that arrive with the tested e-mails.

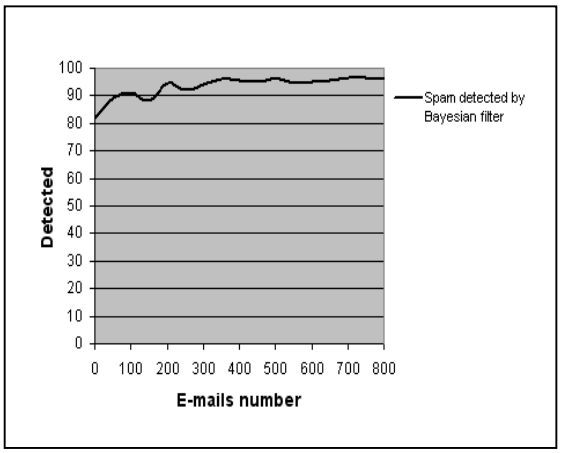


**Fig.8. Spam Detected by the Blocking System.**

Figure (9) examines the effect of four measures TN, TP, FN, and FP on the tested corpus. It reveals that TN and TP increase as more messages are tested whereas FN and FP decrease as the filter has learned more.

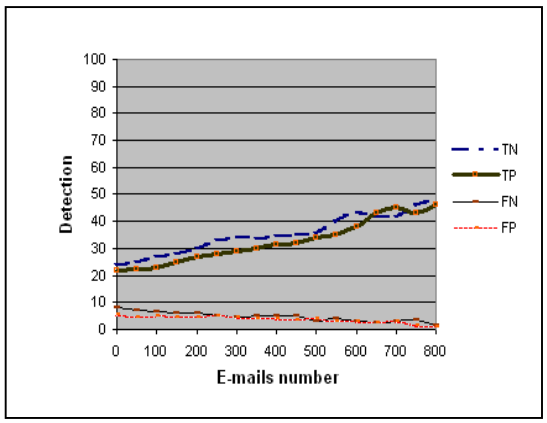


**Fig.9. Percentages of Spam &Legitimate Classification.**

The recent results of the e-mail filter have been used to construct table (1) and figure (10) to summarize the classification results.

**Table 1,**
**Classified E-mails by the Proposed Filter**

| Classification | TN L→L | TP S→S | FN S→L | FP L→S |
|---|---|---|---|---|
| E-mail number | 388 | 369 | 32 | 11 |

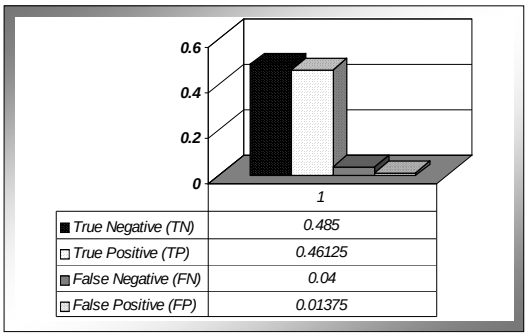


**Fig.10. Percentages of Filter Classification.**

Table (2) shows the results of classification with three values of $\lambda$. As can be seen from the table *WErr* is acceptable low, and it is noticed that with increasing $\lambda$ value the *WErr* decreases. That means the filter has a good low cost resulted from L→S error type. Also as can be observed that the smallest value of $\lambda$ has TCR $\cong$ 9.3, which indicates the high performance of the filter.

**Table 2,**
**Performance Evaluation Results of the Proposed Filter with Three Values of $\lambda$**

| Description | $\lambda$ | e-mail No. | *WAcc* | *WErr* | SP | SR | TCR |
|---|---|---|---|---|---|---|---|
| E-mails evaluation performance | 1 | 800 | 0.946 | 0.054 | 0.97 | 0.92 | 9.302 |
| | 9 | | 0.965 | 0.035 | | | 3.053 |
| | 999 | | 0.969 | 0.030 | | | 0.036 |

## 8. Conclusions

Based on the research undertaken, the analysis and design processes of the filter and the implementation of it, several key results have been identified:

- The proposed filter is an aggregation of whitelist, blacklist in addition to the Bayesian filter. This has been contributed in increasing filter efficiency, decreasing false positive, and reduces classification time.
- Bayesian filtering includes the elements of changeability. That is it is constantly self-updating which makes it very difficult for spammers to circumvent.
- The Bayesian method is multilingual. So, it is efficiently adapted in this work to manage English, Arabic, or mixed linguistic messages.
- The proposed solution is a client side-filtering approach that adds convenience to the user. This allows the users to custom the filter based on their interests and needs.
- A number of performance measurements have been carried out. The results ensure performance of the filter and its efficiency.

## 9. References

[1] Lorenzo Lazzari, Marco Mari and Agostino Poggi, "CAFÉ– Collaborative Agents for Filtering E-mails", Proceedings of the 14th IEEE International Workshops on Enabling Technologies (WETICE'05), 2005.

[2] Xiao-Lin Wang and Ian Cloete, "Learning to Classify E-mail: A Survey", Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, pp. 5716-5719, August 2005.

[3] Yang Li, Binxing Fang, Li Guo and Shen Wang, "Research of a Novel Anti-Spam Technique Based on Usere's Feedbach and Improved Naïve Bayesian Approach", 2006.

[4] Calto Pu, Steve Webb, Oleg Kolesnikov, Wenke Lee, and Richard Lipton, "Towards the Integration of Diverse Spam filtering Techniques", Proceedings of the 22nd International Conference on Data Engineering Workshops (ICDW"06), 2006.

[5] Takamichi Saito, "Anti-Spam System: Another Way of Preventing Spam", Proceeding of the 16th International Workshop on Database Systems Applications (DEXA"05), 2005.

[6] Paul Heymann, Georgia Koutrika, and Hector Garcia-Molina, "Fighting Spam on Social Web Sites: A Survey of Approaches and Future Challenges", IEEE Internet Computing, November 2007.

[7] Behrouz A. Forouzan, Data Communication and Networking, McGraw-Hill, 2007.

[8] Fred Halsall, Computer Networking and the Internet, Addison-Wesley, 2005.

[9] Kevin Johnson, Internet Email Protocols, Developers Guide, Addison-Wesley Longman Inc., 2000.

[10] A. McCallum and K. Nigam, "A Comparison of Event Models for Naïve Bayes Text Classification," Journal on Machine Learning Research 3, pp. 1265-1287, 2003.

[11] Yeates D., and Cadle J., Project Management for Information Systems, 2nd edition London: Finintial Times Management.

[12] Wen-Feng Hsiao, Te-Ming Chang, and Guo-Hsin Hu" A Cluster-based Approach to Filtering Spam under Skewed Class Distributions", Proceedings of the 40th Hawaii International Conference on System Science, pp.1-7, 2007.

[13] I. Androutspoulos, G.Sakkis, C.D. Spyropoulos, and P. Stamatopoulos, "Learning to Filter Spam E-Mails: A comparison of a Naïve Bayesian and a Memory-Based Approach", Proceedings of the Workshop: Machine Learning and Textual Information Access, pp.1-13, 2000.

# تصميم مرشح Bayesian لرسائل الدعاية يعتمد هندسة البرامجيات

**ممتاز محمد علي المختار**
*كلية هندسة المعلومات/ جامعة النهرين*

**الخلاصة**

الانتشار االسريع و توفر السهل لخدمة البريد الاكتروني المجاني جعلا منه وسطا مختارا لارسال بريد الاعلانات الغير مرغوبة و بريد الدعاية بشكل عام. هذه الرسائل، والمعروفة بالبريد التافه او (spam) مشكلة متزايدة لكل من المستعملين و مزودي خدمة الانترنت (ISP). يقدم البحث حلا لاحدى جوانب مشكلة رسائل الدعاية (spam) من خلال تطوير مرشح ملائم لبريد المستفيد (e-mail client). المرشح المقترح يتكون من ثلاثة اجزاء تعمل معا: القائمة البيضاء (Whitelist)، القائمة السوداء (Blacklist)، و مرشح Bayesian. يسمح مرشح القائمة البيضاء باستقبال الرسائل البريدية من عناوين معروفة للمستفيد. بينما يمنع مرشح القائمة السوداء استقبال الرسائل البريدية من عناوين عرفت بارسالها لرسائل الدعاية. يعتمد مرشح Bayesian في تقديراته على محتوى الرسائل ويرشح هذه الرسائل نسبة ال معيار (threshold) محدد سلفا.

تم بناء قواعد البيانات المطلوبة بشكل جداول تخزن في خادم ال SQL. المرشح المقترح للمستفيد يمكن ان يصل الى قواعد البيانات هذه بشكل شفاف لكي يتمكن من تنفيذ الترشيح المطلوب. النظام المقترح يتعامل مع رسائل الدعاية التي تكتب في كلتا اللغتين العربية و الانكليزية و الذي يعتبر امرا هاما للمستفيدين في منطقتنا.

تم اعتماد مبادئ هندسة البرامجيات خلال تصميم النظام مما يجعل النظام اقل عرضة للاخطاء وادامته اسهل. خطوات التصميم نفذت باستحدام نموذج Waterfall وبرامجيات ASCENT. تم تطوير واجهة للمستفيد سهلة الاستخدام للحصول على مزايا المرشح المقترح. تم استخدام بيئة Visual Basic 6 لبناء النظام كما استخدم SQL Server لبناء وتنفيذ قواعد البيانات المطلوبة.

تم استخدام عدد من مقاييس الاداء و استحصال النتائج التجريبية مع مجموعة من البريد المجموع لتقييم الاداء للمرشح المقترح واثبات كفائته.