

available at www.sciencedirect.comjournal homepage: www.elsevier.com/locate/cose

**Computers
&
Security**



Hybrid spam filtering for mobile communication

Ji Won Yoon^a, Hyoungshick Kim^{b,*}, Jun Ho Huh^c

^a Statistics Department, Trinity College Dublin, Ireland

^b Computer Laboratory, University of Cambridge, Cambridge, UK

^c Computing Laboratory, University of Oxford, UK

ARTICLE INFO

Article history:

Received 22 August 2009

Received in revised form

17 October 2009

Accepted 6 November 2009

Keywords:

Spam SMS messages

Hybrid

Content-based filtering

Challenge-response

Threshold sensitivity problem

ABSTRACT

Spam messages are an increasing threat to mobile communication. Several mitigation techniques have been proposed, including white and black listing, challenge-response and content-based filtering. However, none are perfect and it makes sense to use a combination rather than just one. We propose an anti-spam framework based on the hybrid of *content-based filtering* and *challenge-response*. A message, that has been classified as *uncertain* through content-based filtering, is checked further by sending a challenge to the message sender. An automated spam generator is unlikely to send back a correct response, in which case, the message is classified as spam.

Our simulation results show the trade-off between the *accuracy* of anti-spam classifiers and the incurring *traffic overhead*, and demonstrate that our hybrid framework is capable of achieving high accuracy regardless of the content-based filtering algorithm being used.

© 2009 Elsevier Ltd. All rights reserved.

1. Introduction

Short Message Service (SMS) and Multimedia Messaging Service (MMS) are a popular means of mobile communication. Texting costs have decreased continuously over the years (to an extent of free texting) whereas the bandwidth for communication has increased dramatically. Such trends have attracted a large number of phishing and spamming attacks using SMS messages. In particular, spam containing pornographic or promotive materials are an emerging phenomenon and they have caused a significant level of inconvenience for users. These are now prevalent in Korea, Japan and China and prone to spread across countries where mobile communication is popular. Statistics for 2008 (He et al., 2008) show that a user in China, on average, receives 8.29 SMS spam per week.

Much of the existing research into anti-spam solutions, however, has focused on spam emails. Some of the popular methods include white and black listing, digital signature, postage control, address management, collaborative and

content-based filtering (Healy et al., 2005; Metsis et al., 2006; Bratko et al., 2006; Cormack et al., 2007; Dwork et al., 2003; Hall, 1998; Golbeck and Hendler, 2004; Androutsopoulos et al., 2000). Different characteristics between emails and SMS messages make it harder for one to apply such approaches directly in mobile networks and analyze the results (Deng and Peng, 2006). For example, the extra traffic required to perform challenge-response needs to be minimized (or needs to be compensated for) as it is more expensive to use the bandwidth in mobile networks. Also, applying content-based filtering methods to SMS messages is a challenging task since a mobile text message — containing only a small text and phone number — is relatively shorter in length and contains less structured fields compared to an email. With emails, additional fields like attachments, links, and images are commonly used for detecting spam. However, these are not available in SMS messages to construct filtering rules that are as effective as ones used for emails. Due to various drawbacks associated with challenge-response and content-based

* Corresponding author.

filtering, it would make more sense to use a *combination* rather than just relying on one.

In this paper, we propose a spam filtering framework based on combination of these two methods and demonstrate that our combined approach can be more effective and efficient in handling spam SMS messages. Using the content-based filtering approach, obvious spam are filtered first to reduce the number of messages subject to challenge-response; the challenge-response protocol then classifies machine-generated spam with high accuracy. By combining the content filtering algorithm with the challenge-response scheme, we show that, ultimately, high accuracy and low message traffic can be achieved simultaneously. We also describe four challenge-response protocols based on ‘Completely Automatic Public Test to tell Computer and Humans Apart’ (CAPTCHA). Even though many researchers have discussed CAPTCHA based challenge-response protocols (Roman et al., 2006; Shirali-Shahreza and Movaghar, 2008; He et al., 2008), their protocols do not consider the cryptographic details. We extend these protocols for formal verification under a security threat model. Moreover, our simulation results (see Section 4) show that this hybrid approach is capable of controlling high-volume spam and traffic usage.

The remainder of the paper is organized as follows. Section 2 discusses the related work. Section 3 describes the hybrid filtering framework. Section 4 evaluates the performance of the proposed framework based on two measures: traffic usage and accuracy. Finally, Section 5 discusses the contribution of this paper and entails the remaining work.

2. Related work

Content-based filtering solutions have been proved to be effective against emails (Androustopoulos et al., 2000; Metsis et al., 2006; Bratko et al., 2006), which are typically larger in size compared to SMS messages. Abbreviations and acronyms are used more frequently in SMS messages and they increase the level of ambiguity. This makes it difficult to adopt traditional spam filters without any modification. Healy et al. (2005) discuss the problems of performing spam classification on short messages by comparing the performance of the well-known K-Nearest-Neighbor (KNN), Support Vector Machines (SVM), and Naive Bayes classifiers. They conclude that, for short messages, the SVM and Naive Bayes classifiers substantially outperform the KNN classifier, and this contrasts with their previous results obtained for longer emails. Hidalgo et al. (2006) also carried out content filtering experiments with English and Spanish spam SMS corpora to prove that Bayesian filtering methods are still effective against spam SMS messages. Deng and Peng (2006) designed a distributed, content-based filtering method that considers other SMS message characteristics such as its length, which is usually longer than that of a *ham* (normal message).

One of the drawbacks of existing solutions, however, is that they often look for topical terms or phrases such as ‘free’ or ‘viagra’ to identify spam messages. In consequence, some of the legitimate SMS messages that contain such black listed words can be mistakenly classified as spam. This could

happen more frequently with SMS messages than with emails due to their smaller size and simpler content. Moreover, adaptive schemes as such are fundamentally weak against innovative attacks where strategies constantly evolve to manipulate classification rules. Filtering alone will not be sufficient to detect spam.

Many anti-spam solutions (He et al., 2008; Shirali-Shahreza and Movaghar, 2008) have been suggested based on a challenge-response protocol. A message sender needs to prove that they are a human user sender by answering the challenge message (e.g. through a web interface) before their message is forwarded to the recipient. The senders authenticate themselves as a human user by answering a simple Turing test which a machine cannot easily understand. The protocol, however, has often been criticized for extra user interaction and traffic used. There might also be a significant overhead in storing and managing challenge messages. Roman et al. (2006) have introduced a pre-challenge method to overcome these problems. Their method assumes that each user has a challenge associated with their email address. Hence, the email sender can instantly access the recipient’s challenge, and send the response together with the email. Their security model is undermined, however, when the response is exposed to an adversary.

He et al. (2008) proposed a framework which combines white/black listing and challenge-response methods. However, their work does not consider the necessary security, performance, and cost implications of using such a protocol in detail.

3. A hybrid framework

This section describes our hybrid approach. SMS messages are first classified into three different regions using the content-based filtering method: *ham*, *uncertain* and *spam*. Considering that the filtering method is not suitable for dealing with *uncertain* messages, the challenge-response method is then used to further classify the uncertain messages into *ham* and *spam* regions. In practice, the majority of spam messages are generated by machines. Therefore, a human verification mechanism — in the form of challenge-response — is used to detect whether an uncertain message falls into the *ham* or *spam* region.

Fig. 1 shows a high level overview of three major stakeholders: the message sender, message center, and recipient. The message center (owned by the mobile operator) sends a challenge query to check whether the sender is a human or machine. The sender responds by answering the query and the message center compares the returned value against the known correct value. If the values match, the message is classified as *ham*, otherwise, it is classified as *spam*. We are interested in further classification of this uncertain region.

We would suggest that the message center should be given the full responsibility of running our framework for the following reasons:

- to reduce the traffic usage by filtering spam messages at the earliest possible stage; that is, before forwarding them to the recipient.

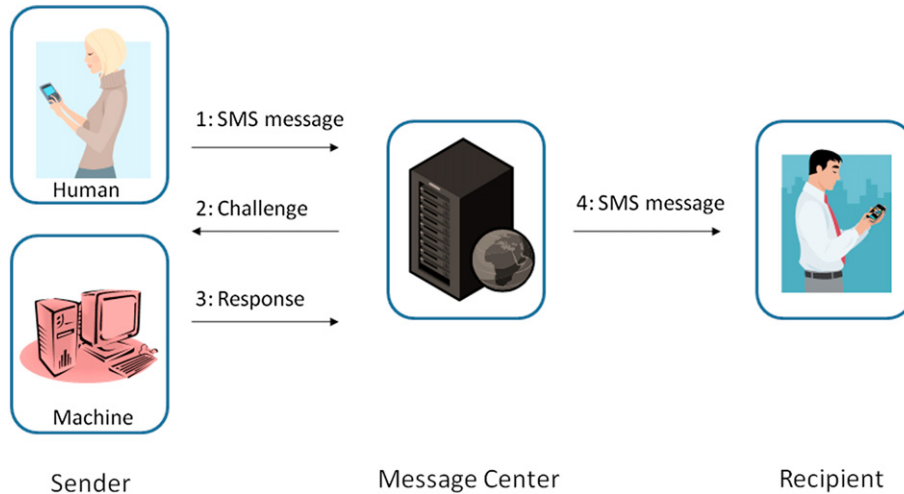


Fig. 1 – Hybrid spam filtering overview.

- by using the challenge-response protocol, the message center will be able to collect a large amount of sample data in real time; these can be used to develop highly effective classifiers and continuously improve the performance of filtering algorithms.
- it would be difficult to install and maintain a homogeneous anti-spam software on all mobile devices; instead we rely on one solution deployed in the message center.

In practice, however, it is possible that the operator of the message center would allow certain companies to send spam messages to its users for a payment. Our work assumes that the operator always works in the best interest of the user and will only allow such messages to go through if the user has agreed to receive messages from these companies. We imagine that the message center holds the user's white list of 'interesting companies' and only forwards messages from the listed companies.

3.1. Introducing the uncertain region

If we assume there are only two regions — ham and spam — the content-based filter will use binary classification. Suppose that we have a probabilistic model for the anti-spam classifier as a posterior distribution $Pr(c = \text{ham}|y)$. This is the probability that a message falls into the ham region: c and y denote realization of random variables for a class and message, respectively. The odd ratio of the posterior is used to obtain a measurable classification by $O_{\text{post}} = Pr(c = \text{ham}|y) / Pr(c = \text{spam}|y)$. If $O_{\text{post}} > 1$, a message is classified as ham, otherwise, as spam. Alternatively, we can simply use a threshold based approach in the posterior distribution. If $Pr(c = \text{ham}|y)$ is closer to one, a message is likely to be ham; if closer to zero, it is likely to be spam. Let $\bar{c} = f(y, h)$ be the content-based filter where \bar{c} and h are the output and given threshold, respectively. This filter would work with the following rules:

$$\bar{c} = f(y, h) = \begin{cases} \text{ham} & \text{if } Pr(c = \text{ham}|y) \geq h \\ \text{spam} & \text{if } Pr(c = \text{ham}|y) < h \end{cases} \quad (1)$$

This separates ham from spam (the odd ratio approach is a special case where $h = 0.5$). The main problem with this approach is finding a proper threshold: because the threshold for ground truth \tilde{h} is unknown, there are two possible cases as shown in Fig. 2(a).

If h is higher than \tilde{h} , some of the ham in region A could be classified as spam. If h is lower than \tilde{h} , some of the spam in region B could bypass the content-based filter and reach the recipients. Such a threshold problem will always be present in classification: it is almost impossible to find the underlying \tilde{h} , and the anti-spam software companies are likely to use strategies based on their own experiences. In order to minimise the false negatives (i.e. ham being classified as spam and not reaching the recipient) in mobile networks, binary methods tend to be configured with less sensitivity with regards to detecting spam. They would rather mistakenly forward spam than prevent any legitimate message from reaching the recipient. We believe that these problems can be resolved by introducing an uncertain region with two thresholds (see Fig. 2b). These can be implemented as the upper and lower boundaries of a traditional threshold system. As a result, we now have three labels: spam, uncertain area, and ham — the focus is on the uncertain area. Spam and ham regions are classified as in the traditional system. Only the messages that fall into the uncertain area are checked further using the challenge-response protocol. The next section describes our proposed protocols in detail.

3.2. Challenge-response protocols

First, we assume that there is a Turing test available with a low probability of producing false positives and false negatives. CAPTCHA is a commonly used one — it generates pattern matching problems for which a human can easily recognize and solve, whereas a machine cannot. An automated program that generates thousands of spam will not be capable of answering a CAPTCHA based challenge, which could be a graphical image containing a faint typeface. If the response is correct, there is a high probability that the sender is a human. CAPTCHA can be designed in different media

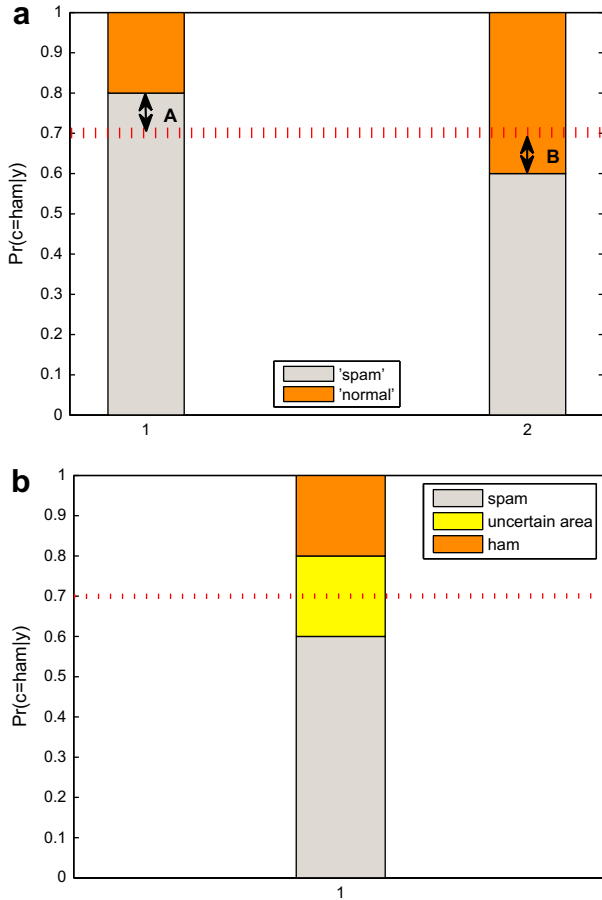


Fig. 2 – (a) Two possible cases: $h > \hat{h}$ (case 1) and $h < \hat{h}$ (case 2) for a given ground truth \hat{h} (red dot line) and (b) modified classification embedding uncertain area given a ground truth \hat{h} (red dot line).

forms such as an image, an audio file or a text (von Ahn et al., 2008). Their implementation details, however, are beyond the scope of this paper.

A number of challenge-response protocols have already been proposed (He et al., 2008; Shirali-Shahreza and Movaghar, 2008). Although, these focus only on the implementation issues without considering the security model and cryptographic details. We define our own security models and describe a number of possible protocols in line with them. There are several issues we need to consider before designing the protocols:

- when we are dealing with spam, message authentication and integrity are important, whereas confidentiality is not.¹
- SMS messages are usually unencrypted and unsigned; hence, it is possible to tamper with them during transmission.
- security properties of the communication channel between the message center and the sender need to be defined; this channel might or might not be an authenticated one.

¹ The adversary's goal is to deliver spam messages to the recipient.

- managing session information between all trusted pairs for challenge-response would impose huge storage overhead on the message center; there might be more than one message center sharing this information, and it might or might not be stored in the center.

Mindful of these security and scalability issues, we proposed four different protocols: protocols 3 and 4 assume an authenticated channel, whereas protocols 1 and 2 do not; protocols 1 and 3 assume that the message center manages the session information, whereas the others do not.

3.2.1. Protocol notations

Standard engineering notations (Burrows et al., 1989) were used in describing the protocols. In a protocol that is used by A and B, “ $A \rightarrow B : X$ ” implies that A sends message X to B. The symbols S and R represent the Sender and Recipient, respectively. M represents the Message center, T a Timestamp, N a Nonce, K a Key and K^{-1} its inverse. In a symmetric cryptosystem such as AES, K and K^{-1} are always equal. A Plain SMS message encrypted with K is represented as $\{P\}_K$. H is a one-way hash function. The subscript m in K_m implies that K_m is M's public key. Additionally, ms in K_{ms} shows that K_{ms} is intended for communication between M and S.

The sender's ability to send a correct response depends on their competence to interpret the key, K_c^{-1} . An unauthorized sender (e.g. a program sending spam) will not be able to interpret and figure out K_c^{-1} — this key serves to identify machine-generated spam. For simplicity, encryption algorithms were not considered in the protocols.

3.2.2. Protocols

In protocol 1, the message center (M) maintains the session information.

[Protocol 1]

(M1) $S \rightarrow M : S, R, P$

(M2) $M \rightarrow S : M, S, \{K_{ms}\}_{K_c}, \{H(S, R, P), N\}_{K_{ms}}$

(M3) $S \rightarrow M : S, M, \{H(S, R, P), N + 1\}_{K_{ms}}$

Before sending message 1, S stores R and P to prevent message modification attacks. After receiving message 1, M generates K_{ms} and stores (S, R, P, K_{ms} , N) as the session information. K_{ms} is protected with K_c . An image CAPTCHA would be one way of protecting K_{ms} against spam programs. After receiving message 2, S decrypts $\{K_{ms}\}_{K_c}$ by answering the challenge (their ability to interpret K_c^{-1}). S then decrypts $H(S, R, P)$ and N using K_{ms} . S compares $H(S, R, P)$ against the previously stored values. S terminates the protocol if these values do not match; otherwise, S generates $\{H(S, R, P), N + 1\}_{K_{ms}}$ by K_{ms} and sends it to M. After receiving message 3, M verifies $\{H(S, R, P), N + 1\}_{K_{ms}}$. If it is valid, M forwards the stored message (S, R, P) to R. Finally, M deletes the session information. The proof of this protocol is presented in Appendix A.

The users could become frustrated, however, if they receive too many challenge messages. We use a timestamp (T) to solve this problem. After receiving message 3, M maintains a session information (S, R, P, K_{ms} , T) between S and R for a given time interval. M checks the validity of K_{ms} using the session information and a policy that defines the lifetime

of K_{ms} . This is also effective for detecting and avoiding any replay attacks.

The main drawback of this protocol is that M has to bear the huge overhead of maintaining the session information. We describe another protocol which solves this issue by using authorized tokens instead:

[Protocol 2]

(M1) $S \rightarrow M : S, R, P$

(M2) $M \rightarrow S : M, S, \{K_{ms}\}_{K_c}, \{H(S, R, P)\}_{K_{ms}}, \{K_{ms}, H(S, R), T\}_{K_m^{-1}}$

(M3) $S \rightarrow M : S, R, \{P\}_{K_{ms}}, \{K_{ms}, H(S, R), T\}_{K_m^{-1}}$

The main difference is the use of $\{K_{ms}, H(S, R), T\}_{K_m^{-1}}$ (which can only be generated by M) as the authorization token for verifying a response. M checks whether S is authorized by looking at $\{K_{ms}, H(S, R), T\}_{K_m^{-1}}$. Using this token, S can just send message 3 alone, including a new text (P'), within the lifetime of T :

(M1) $S \rightarrow M : S, R, \{P'\}_{K_{ms}}, \{K_{ms}, H(S, R), T\}_{K_m^{-1}}$

In these protocols, however, S cannot verify the authenticity of the challenge message. Before describing the next two protocols which aim to solve this problem, we make an assumption that there is an authenticated channel between M to S , and M 's public key (K_m) is securely installed on a mobile device owned by S (perhaps during the process of manufacturing). We describe the following protocols based on this assumption:

[Protocol 3]

(M1) $S \rightarrow M : S, R, P$

(M2) $M \rightarrow S : M, S, \{\{K_{ms}\}_{K_c}, N\}_{K_m^{-1}}$

(M3) $S \rightarrow M : S, R, \{N + 1\}_{K_{ms}}$

In protocol 3, M maintains the session information, (S, R, P, K_{ms}, N). When message 2 arrives, S verifies the signature on $\{\{K_{ms}\}_{K_c}, N\}_{K_m^{-1}}$. S does not respond if the signature is invalid.

[Protocol 4]

(M1) $S \rightarrow M : S, R, P$

(M2) $M \rightarrow S : M, S, \{\{K_{ms}\}_{K_c}, H(S, R), T\}_{K_m^{-1}}, \{P\}_{K_m^{-1}}$

(M3) $S \rightarrow M : S, R, \{P\}_{K_{ms}}, \{\{K_{ms}\}_{K_c}, H(S, R), T\}_{K_m^{-1}}$

Protocol 4 uses $\{\{K_{ms}\}_{K_c}, H(S, R), T\}_{K_m^{-1}}$ as the authorized token. Our protocols are likely to be compatible with existing devices since the majority already have built-in encryption and hash functions.

3.3. Observations

3.3.1. Upgrading protocols

A message is always sent to the message center of the contracted operator first. If the message is directed at someone contracted to a different operator, it is forwarded to another message center before reaching the recipient's handset (Enck et al., 2005). This means if one of the message centers decides not to use our framework, all uncertain texts delivered via that center would bypass the content-based filter. It would be the weakest point (and the only route needed) for an attack.

Hence, all existing message centers would have to support the new protocol. While this is a large change and a challenging one, operator-sponsored forums like OMTP (Open Mobile Terminal Platform), are working with key mobile operators to unify and recommend mobile terminal requirements (Rogers, 2007). With the increasing number of spam texts, it seems likely that the ability to filter machine-generated uncertain texts will persuade operators into upgrading their systems.

3.3.2. Performance degradation

If there are too many messages subject to challenge-response, its overhead will dominate. For example, sending an image CAPTCHA is a huge overhead to authenticate a 100 character SMS message. Future work may look at adding a 'bypass' to the hybrid, so that a message originating from a verified sender can be automatically treated as ham without having to go through the spam filtering process.

For instance, the message center could manage the recipient's white list of acceptable phone numbers — typically, through synchronization with the recipient's contact list. Since the message center has secure access to the message sender details (including the phone number), it can first check to see if the sender's phone number is included in the recipient's white list. If it is a listed number, the message can be treated as ham and forwarded to the recipient; if not, the message can go through the spam filtering process. As the uncertain region becomes smaller, we expect the performance of our framework to improve.

3.3.3. Usability issues

Adapting CAPTCHA methods will have implications on usability. A mobile device might not have the capability to display an image CAPTCHA to a readable standard; also a mobile user might find it difficult to verify an audio CAPTCHA due to background noise — hence, it is important to set up user-friendly CAPTCHA methods.

Different approaches for generating user-friendly CAPTCHA messages have been discussed by various researchers (Leveraging the CAPTCHA Problem, 2005; Yan and El Ahmad, 2008). Chow et al. (2008) proposed a new CAPTCHA technique that minimizes the level of user frustration and facilitate the use of CAPTCHA on mobile devices. Their technique is well suited for keyboard-less mobile devices.

4. Evaluation

Fig. 3 demonstrates a basic SMS deployment architecture and its wireless network components (Prieto et al., 2004): the Home Location Register (HLR), Mobile Switching Center (MSC), SMS Gateway (SMSG), and SMS Center (SMSC) are such components. These are interconnected as shown in Fig. 3.

4.1. Description of datasets

In order to measure the performance of our framework, we generated synthetic datasets. Suppose that there are N sent messages (we set $N = 5000$). We use p and q to show the ham to spam proportion where $p + q = 1$, and p and q are non-negative numbers (in reality, different operators will have different

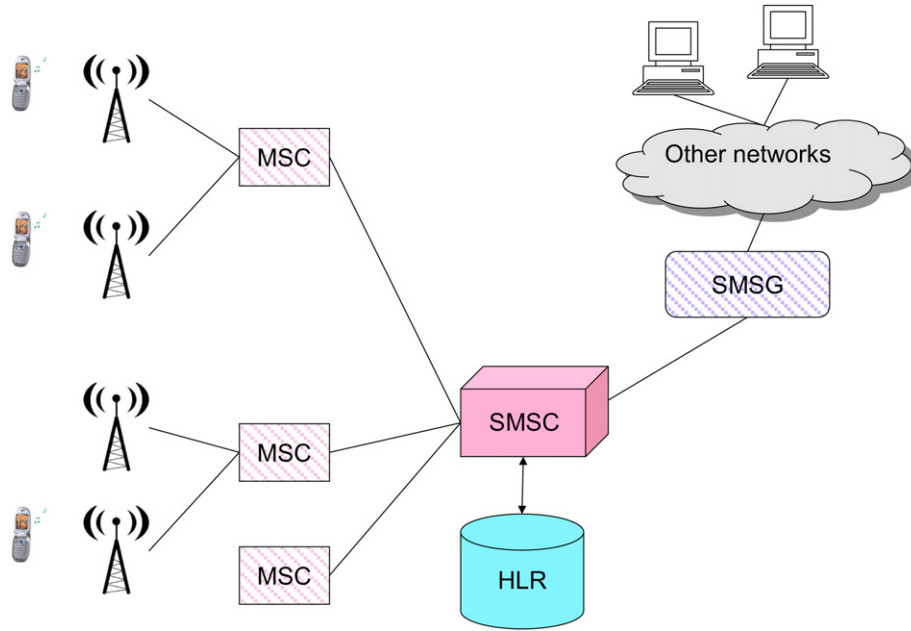


Fig. 3 – SMS deployment architecture.

proportions). Let κ be a random variable generated from an existing filtering method, given an observed data y : $\kappa = Pr(c = \text{ham}|y)$. For an artificial dataset, we build a mixture model given by

$$p(\kappa|\lambda) = p(\kappa|c = \text{ham}, \lambda)p(c = \text{ham}|\lambda) + p(\kappa|c = \text{spam}, \lambda)p(c = \text{spam}|\lambda) \quad (2)$$

where λ denotes a set of hyper-parameters which control parameters. Since c can only be 0 (spam) or 1 (ham), we assume c_i is generated from Bernoulli distribution with hyper-parameters p and q . Thus, we have:

$$c \sim p(c|\lambda) = \text{Bernoulli}(c; p) = p^c(1-p)^{1-c} = p^c q^{1-c}$$

After classifying the i th sample message, we generate the expected probability (this is the filtering output):

$$\kappa \sim p(\kappa|c, \lambda) = \begin{cases} p(\kappa|c = \text{ham}, \lambda) = \mathcal{B}(\kappa; \alpha_1, \beta_1) \\ p(\kappa|c = \text{spam}, \lambda) = \mathcal{B}(\kappa; \alpha_0, \beta_0) \end{cases}$$

Beta distribution (\mathcal{B}) was used here: κ denotes the probability of ham classified from the existing filtering method, so the random variable lies between 0 and 1; κ can be designed by beta distribution to continuously generate samples between 0 and 1. Both thresholds (h_1 and h_2) vary between 0 and 1 by 1/30. In practice, the hyper-parameters, α_0 , β_0 , α_1 and β_1 , are obtained by the means and variances of spam and ham respectively; this is given by:

$$\alpha_i = -\mu_i + \frac{\mu_i^2(1-\mu_i)}{\sigma_i^2} \quad (3)$$

$$\beta_i = \alpha_i \left(\frac{1}{\mu_i} - 1 \right)$$

where μ_i and σ_i are means and standard deviation of the costs/likelihood of spams ($i = 0$) and hams ($i = 1$).

For an example of this paper, we use $\mu_0 = 0.3750$, $\sigma_0 = 0.7143$, $\mu_1 = 0.1614$ and $\sigma_1 = 0.1597$ and then the hyper-parameters of beta distribution are set as follows: $\alpha_0 = 3$, $\beta_0 = 5$, $\alpha_1 = 5$, $\beta_1 = 2$. Also, we built an artificial dataset based on a Spanish database (Hidalgo et al., 2006) which shows the proportion of spam as 14.57% and ham as 85.32% — that is, $q = 0.1457$ and $p = 0.8532$. Fig. 4 shows the distribution of generated data: in graph (a), the red crosses represent ham and blue circles represent spam; the same colouring scheme is used in graph (b). These graphs show that there is a large amount of overlapping labels between 0.2 and 0.8. This overlapping section is considered as the *uncertain region*. Since the challenge-response protocol is not perfect, some spam will bypass the protocol with correct responses, and some ham will be filtered mistakenly with incorrect responses. To model this imperfection, we use e_1 and e_2 to represent the ratios of False Positives (FP) and False Negatives (FN) in the uncertain region.

In addition, Sections 4.4 and 4.5 use a wide range of randomly generated parameters to demonstrate how performance is affected in various environments: Section 4.4 studies performance with varying ham and spam proportions; Section 4.5 uses fixed proportions of ham and spam ($q = 0.1457$ and $p = 0.8532$), and varies other parameters to study how performance changes.

4.2. Traffic usage comparison

We simulated the traffic usage using the variable thresholds and analyzed the results. Our framework considers several stakeholders (see Fig. 3): the message Sender (S), message Receiver (R), message center (either MSC or SMSG), and other network components (SMSC, HLR).

First, we calculated the traffic used by an existing filtering method. In practice, the size of each message in the protocol will be different. For instance, the size of the challenge

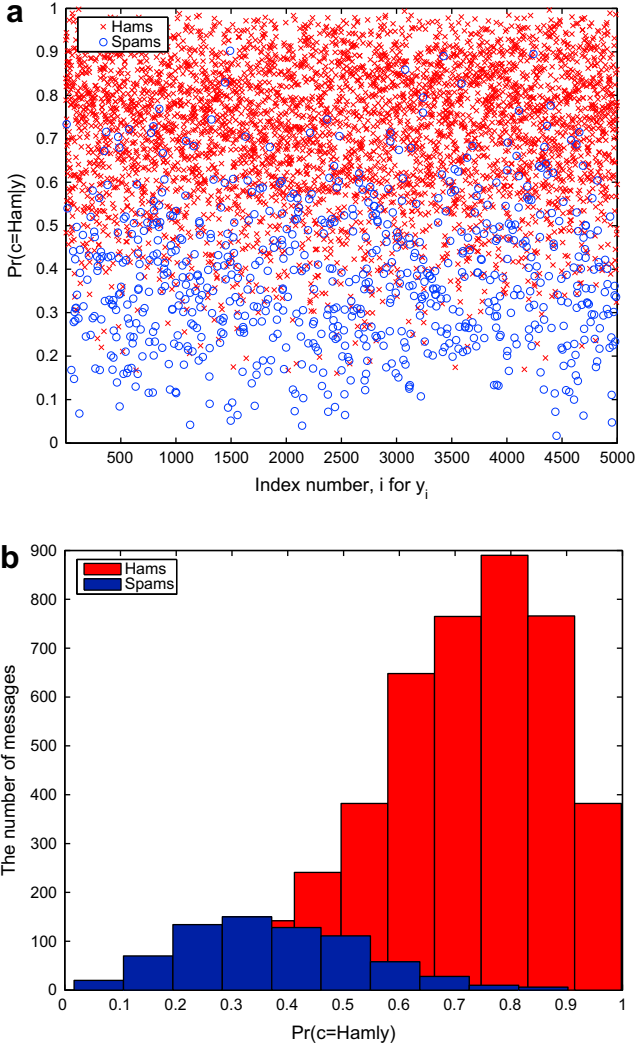


Fig. 4 - Displaying $\kappa = \Pr(c = ham|y)$ for N messages: spam (14.57%) and ham (85.32%). (a) N messages and (b) distribution.

message will be greater than other text messages since it would include a CAPTCHA image. For simplicity, however, we assume that all messages have the same size.

Only the messages with filtering probabilities higher than the threshold h reach R via the network components; other

messages are deleted at the message center. Suppose that $y_{\bar{c}=type}^h$ for $type \in \{ham, spam\}$ denotes all messages filtered as $type$ in terms of h , then the total amount of traffic used is the sum of $|y_{\bar{c}=ham}^h| \times 6$, and $|y_{\bar{c}=spam}^h| \times 1$ where $|\cdot|$ represents the cardinality of a set: $N_{FilteringOnly} = |y_{\bar{c}=ham}^h| \times 6 + |y_{\bar{c}=spam}^h| \times 1$. This is because a ham traverses through six different paths ($S \rightarrow MSC/SMSG \rightarrow SMSC \rightarrow HLR \rightarrow SMSC \rightarrow MSC \rightarrow R$) whereas a spam just traverses through one ($S \rightarrow MSC/SMSG$).

In contrast, our hybrid model divides the measurable space into three different areas using two thresholds: h_1 and h_2 . As a result, we have two more parameters to estimate: the traffic used by ham (N_{un}) and spam (N_{us}) in the uncertain region. Let $y_{\bar{c}=type}$ for $type \in \{ham, spam\}$ be a set of messages that have label $type$ as a ground truth.

As Fig. 5 shows, there are four possible pathways:

- in (a), a message classified as ham (using the higher threshold) is sent directly to R via the network components; the number of paths taken is six: $S \rightarrow MSC/SMSG \rightarrow SMSC \rightarrow HLR \rightarrow SMSC \rightarrow MSC \rightarrow R$.
- in (b), a message is in between the higher and the lower thresholds; a correct response is submitted by the sender and the message is classified as ham; the number of paths taken is eight: $S \rightarrow MSC/SMSG \rightarrow S \rightarrow MSC/SMSG \rightarrow SMSC \rightarrow HLR \rightarrow SMSC \rightarrow MSC \rightarrow R$.
- in (c), a message is in between two thresholds again; this time no response is returned and the message is classified as spam; the number of paths taken is two: $S \rightarrow MSC/SMSG \rightarrow S$.
- in (d), a message, classified as spam using the lower threshold, is deleted at the message center (MSC/SMSG); the number of paths taken is one: $S \rightarrow MSC/SMSG$.

The traffic usage is calculated using:

$$\begin{aligned}
 N_n &= |y_{\bar{c}=ham}^{h_2}| \times 6 \\
 N_{un} &= \left| y_{\bar{c}=ham}^{h_1} \cap y_{\bar{c}=spam}^{h_2} \cap y_{\bar{c}=ham} \right| \times (1 - e_1) \times 8 \\
 &\quad + \left| y_{\bar{c}=ham}^{h_1} \cap y_{\bar{c}=spam}^{h_2} \cap y_{\bar{c}=spam} \right| \times e_2 \times 8 \\
 N_{us} &= \left| y_{\bar{c}=ham}^{h_1} \cap y_{\bar{c}=spam}^{h_2} \cap y_{\bar{c}=spam} \right| \times (1 - e_2) \times 2 \\
 &\quad + \left| y_{\bar{c}=ham}^{h_1} \cap y_{\bar{c}=spam}^{h_2} \cap y_{\bar{c}=ham} \right| \times e_1 \times 2 \\
 N_s &= |y_{\bar{c}=spam}^{h_1}| \times 1 \\
 N_{hybrid} &= N_n + N_{un} + N_{us} + N_s
 \end{aligned} \tag{4}$$

where e_1 is a probability that humans fail to respond correctly and e_2 is a probability that spam generated by machines pass

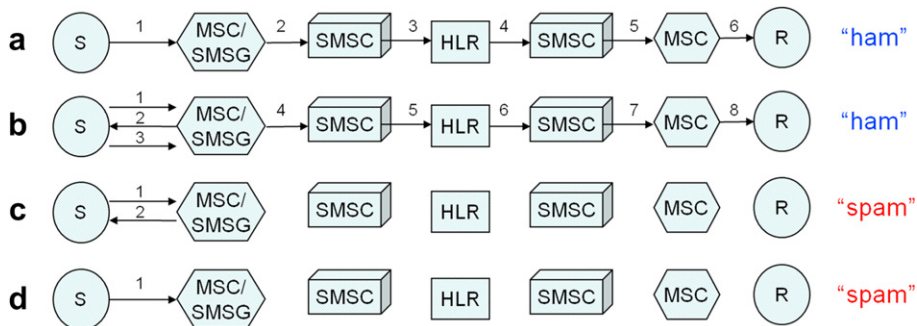


Fig. 5 - Four possible pathways for the hybrid method.

a Turing test. Again, for simplicity, we assume that these probabilities are relatively low and set e_1 and e_2 at 0.02 and 0.01, respectively.

Fig. 6 shows the traffic usage and accuracy with varying thresholds, h_1 and h_2 . We assume that h_1 is smaller than h_2 , and only the right half of the graph is meaningful since the right and left halves of the graph are symmetric. The green plane represents the traffic used by the filtering method alone, and the blue represents the traffic used by our hybrid framework.

In order to show the changes in traffic usage with two varying thresholds, the inner sections of Fig. 6 were explored further in Fig. 7. Graph (a) was plotted with the higher threshold fixed to 0.73333, and with the lower threshold increasing from 0 until it reached this value. The graph shows that the traffic usage decreases as the lower threshold increases. Additionally, the traffic usage ratios of our hybrid approach and the conventional approach (filtering only) are 1.37 (= 4.1/3) and 1.27 (= 3.3/2.6) respectively (at low thresholds 0 and 0.5). From this, we concluded that the amount of traffic used in our approach is roughly 1.3 times greater than that of the conventional approach. We also monitored the traffic usage with the lower threshold fixed to = 0.1, and with the higher threshold increasing from 0.1 to 1 (see graph (b) in Fig. 7). The traffic usage does not change with the filtering-only approach because the lower threshold is the same as h . As the number of messages in the uncertain region increases so does the traffic usage.

4.3. ROC comparison

One of the good measures used in classification is Receiver Operating Characteristic (ROC) curves. We calculated and compared True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) of the underlying classes and the expected ones between the filtering-only method and our hybrid method. Let θ be {TP, TN, FP, FN}. We obtained the proper estimate of θ from the posterior distributions: $p(\theta|h)$ was used for the filtering method and $p(\theta|h_1, h_2)$ for the hybrid method. θ was obtained in the filtering-only method by

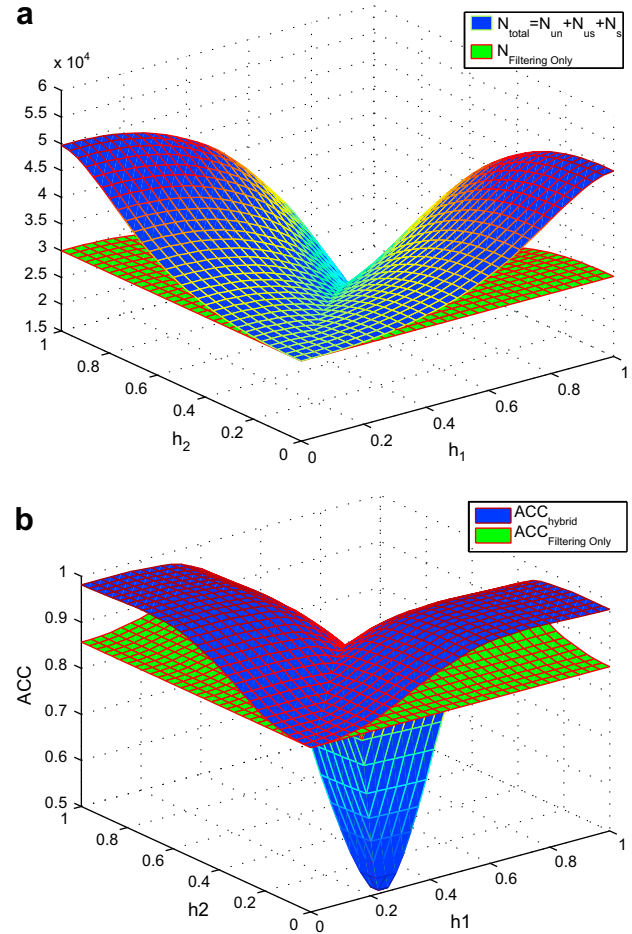


Fig. 6 – 3D view of the (a) traffic usages and (b) accuracy in terms of varying thresholds.

$$\hat{\theta}_{\text{filtering only}} = E(\theta|h, \mathbf{y}) \tag{5}$$

where $E(\cdot|h, \mathbf{y})$ denotes expectation given a threshold. We also obtained θ in the hybrid method by $E(\theta|h_1, h_2, \mathbf{y})$, which denotes expectation given two thresholds. We used marginalized

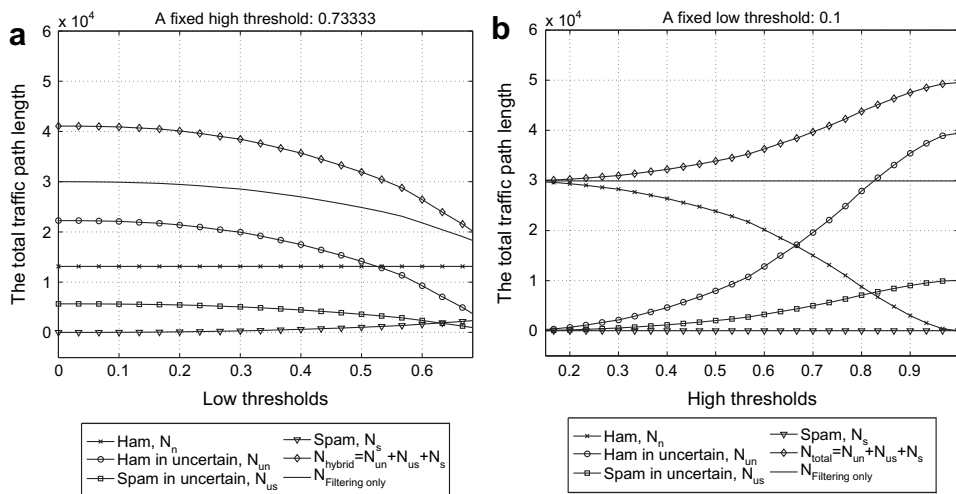


Fig. 7 – Slides of an axis (with fixed threshold). (a) A fixed high threshold and (b) a fixed low threshold.

posterior distribution for the hybrid method since the number of thresholds need to be equal for a fair comparison.

$$\begin{aligned}
 \tilde{\theta}_{\text{hybrid}} &= E(\theta|h_1, \mathbf{y}) \\
 &= \int_{\theta} \theta p(\theta|h_1, \mathbf{y}) d\theta \\
 &= \int_{\theta} \theta \left[\int_{h_2} p(\theta, h_2|h_1, \mathbf{y}) dh_2 \right] d\theta \\
 &= \int_{\theta} \theta \left[\int_{h_2} p(\theta|h_2, h_1, \mathbf{y}) p(h_2|h_1) dh_2 \right] d\theta \\
 &= \int_{\theta} \int_{h_2} \theta p(\theta|h_2, h_1, \mathbf{y}) p(h_2|h_1) dh_2 d\theta \\
 &= \int_{\theta} \int_{h_2} \theta p(\theta|h_2, h_1, \mathbf{y}) d\theta p(h_2|h_1) dh_2 \\
 &\approx \frac{1}{|H_2|} \sum_{h_2 \in H_2} \int_{\theta} \theta p(\theta|h_2, h_1, \mathbf{y}) d\theta \\
 &= \frac{1}{|H_2|} \sum_{h_2 \in H_2} E(\theta|h_1, h_2, \mathbf{y})
 \end{aligned}
 \tag{6}$$

$h_2 \sim p(h_2|h_1)$ and H_2 are a set of samples h_2 . From Eq. (5) and Eq. (6), we plotted an ROC curve with the threshold increasing from 0 to 1 (see Fig. 8); x-and y-axis stand for 1-specificity and sensitivity; these are estimated by

$$\text{specificity} = \frac{TN}{FP + TN} \text{ and } \text{sensitivity} = \frac{TP}{TP + FN} \tag{7}$$

The plain black line is used to show the filtering-only method. The coloured lines with markers are used for the others that use the hybrid method. We have tested four different e_1 s and e_2 s: $e_1 \in \{0.02, 0.04\}$ and $e_2 \in \{0.008, 0.01\}$; these are shown with the coloured lines. This graph shows that our hybrid method has higher performance than the other. The ROC can also be used to generate a summary statistic. One of the common versions is the Area Under the ROC Curve (AUC). The AUC corresponds to the probability of a classifier ranking a randomly chosen positive instance higher than a negative one. The comparison of AUC for all methods is described in Table 1. In this table, the AUCs of all hybrid methods are higher than that of the filtering method. This emphasizes our previous result of the hybrid method having a superior performance. In addition, as the ratios of e_1 and e_2 become smaller, the AUC increases.

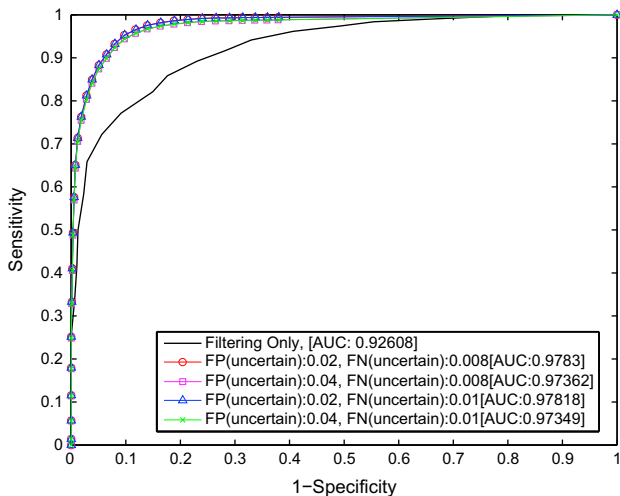


Fig. 8 – ROC curve.

Table 1 – Comparison of AUC.

Method	Ratio (e_1)	Ratio (e_2)	AUC
Filtering only	–	–	0.9261
Hybrid	0.02	0.008	0.9783
Hybrid	0.04	0.008	0.9736
Hybrid	0.02	0.01	0.9782
Hybrid	0.04	0.01	0.9735

4.4. Variant proportion of spam

Previously, in Sections 4.2 and 4.3, the proportions of spam and ham were fixed to 14.57% and 85.32% respectively. In this section we show how the performance is affected when these proportions change.

Table 2 describes a small number of samples from the nine different proportions. Each record has six different columns: proportion of spam (%), lower threshold (h_1), higher threshold (h_2), traffic usage (TU) of N_{hybrid} , ratio ($= N_{\text{hybrid}}/N_{\text{filteringonly}}$), and accuracy ($ACC = (TP + TN)/(P + N)$). It uses three different measures for the performance. If the traffic usage is less, we say the system is lighter and is more economical. The ratio is only close to 1 if the traffic used in the hybrid method is close to the amount used in the other. The accuracy measures the correctness of message classification. We can select practical threshold values for each spam proportion to compare the performance. For instance, threshold values $h_1 = 0.1$ and $h_2 = 0.2$ can be selected in 10% spam proportion to show a reasonable performance of the hybrid method. However, if the system is concerned with achieving high accuracy and not with reducing the traffic usage, $h_1 = 0.1$ and $h_2 = 0.9$ values can be used. In a spam-dominant environment (for spam

Table 2 – Traffic amounts and accuracy of hybrid methods in terms of thresholds.

Proportion of spam	h_1	h_2	TU	Ratio	ACC
10%	0.1	0.2	30,136.4	1.0079	0.9185
	0.1	0.9	47,440.88	1.5867	0.9831
	0.4	0.6	31,683.34	1.1438	0.9527
	0.8	0.9	16,553.18	1.3111	0.3968
20%	0.1	0.2	30,305.68	1.0161	0.8331
	0.1	0.9	47,539.24	1.5939	0.9839
	0.4	0.6	30,522.02	1.1625	0.9415
30%	0.8	0.9	15,569.8	1.3035	0.4693
	0.1	0.2	30,487.38	1.0234	0.7470
	0.1	0.9	47,764.28	1.6034	0.9846
40%	0.4	0.6	29,691.62	1.1843	0.9421
	0.8	0.9	14,176.7	1.2882	0.5312
	0.1	0.2	30,697.42	1.0322	0.6635
	0.1	0.9	47,797.02	1.6072	0.9853
50%	0.4	0.6	28,747.42	1.2053	0.9329
	0.8	0.9	12,850.56	1.2605	0.5963
	0.1	0.2	30,863.82	1.0402	0.5826
	0.1	0.9	47,892.5	1.6142	0.9863
	0.4	0.6	27,396.98	1.2228	0.9275
	0.8	0.9	11,703.38	1.2411	0.6625

proportion of 50%), reasonable threshold values would be $h_1 = 0.4$ and $h_2 = 0.6$. Returning back to the figures for a spam proportion of 10%, $h_1 = 0.1$ and $h_2 = 0.9$ will be selected when accuracy is the most significant factor.

4.5. Different content-based filtering parameters and performance implications

The performance (accuracy and traffic usage) of the proposed hybrid method will vary depending on the characteristics of the content-based filtering method being used. In order to study this further, we simulated the hybrid method with 200 random samples, each representing a different case of the content-based filtering method. The following parameters were considered for each sample:

- μ_0 : the mean of the cost/likelihood of spam
- σ_0 : the standard deviation of the cost/likelihood of spam
- μ_1 : the mean of the cost/likelihood of ham
- σ_1 : the standard deviation of the cost/likelihood of ham
- h_1 : a lower threshold
- h_2 : a higher threshold.

From the results, their accuracy and ratio of traffic usage (hybrid method to conventional filtering-only method) values were plotted in a graph (see Fig. 9). The results were clustered using a well-known clustering technique called the K-means algorithm (Hartigan, 1975) by setting $K=4$ (implying 4 clusters). The samples clustered around the top left region of the graph (assigned with label 1) are regarded as *recommendable cases* since they show high accuracy but with smaller increase in the traffic usage compared to the filtering-only method. The samples clustered around the top right region of the graph (assigned with label 2) show high accuracy but suffer from a large increase in the traffic usage compared to the filtering-only method. This implies that the hybrid method should be used for these samples when

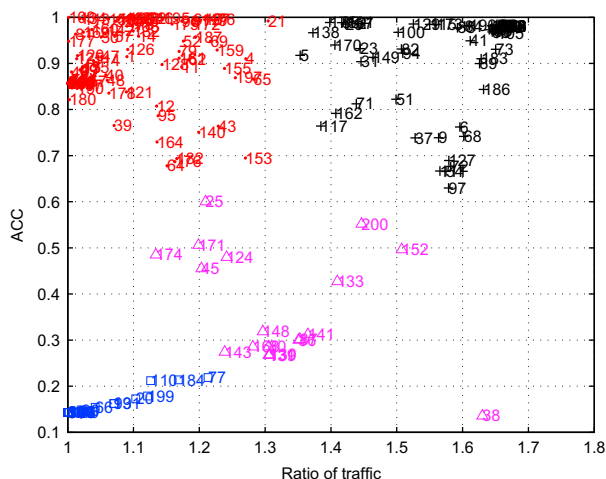


Fig. 9 – Accuracy and traffic usage ratio for 200 random samples — each assigned with one of the four labels: label 1 (red dots, top left), label 2 (black crosses, top right), label 3 (blue squares, bottom left) and label 4 (pink triangles, center).

achieving a high accuracy is considered relatively more important than the resulting increase in the traffic usage. The worst set of samples are those clustered around the bottom left region of the graph (assigned with label 3). Although these show small increase in the traffic usage, the accuracy is also very low. For this reason, the hybrid method is not really suitable for handling such samples. The samples clustered broadly around the center region of the graph (assigned with label 4) are considered better than those assigned with label 3, but worse than those assigned with labels 1 and 2.

More can be observed from an associated set of data, presented in Table 3 (see Appendix B). Each record describes a sample profile.² Different applications and business models will have different preferences for these two ratios. Taking this into consideration, we studied the trends for three representative cases where

1. accuracy is considered twice as important as traffic usage – Fig. 10(a)
2. both are considered as equally important – Fig. 10(b)
3. traffic usage is considered twice as important as accuracy – Fig. 10(c)

The results are shown in Fig. 10. In Fig. 10(a), the samples which satisfy $\Delta_{ACC} > 0.5\Delta_{TU} + 0.5$ are plotted with red dots and these represent good samples. The others, plotted with black 'x's, are classified as relatively bad samples. Similarly, in Fig. 10(b) and (c), good samples (red dots) satisfy $\Delta_{ACC} > \Delta_{TU}$ and $\Delta_{ACC} > 2\Delta_{TU} - 1$, respectively, whereas relatively bad ones (black 'x's) do not.

We also plotted the ratio between the number of good samples and bad samples as derived from these three cases (see Fig. 11). The graph shows that, with the varying importance of these two factors, the ratio of the number of good samples to bad samples changes: as the importance of accuracy (relative to traffic usage) increases, so does the number of good samples, and vice versa.

Figs. 10 and 11 both show that as the traffic usage becomes more important, the number of good samples decreases. Hence, if our hybrid method was to be applied in an environment where traffic usage is considered relatively more important, the content-based filtering parameters as well as the threshold values need to be selected more carefully.

² A sample profile consists of the specific parameter values. Given the space available, we only show 10 representative samples for each label (L). Consider the samples that have been assigned with labels 3 ($L=3$) and 4 ($L=4$). In practice, there is a very low chance for these cases to arise since the means of spam and ham (μ_0 and μ_1) are lower than the lower threshold, and the standard deviations are also relatively small. Therefore, if we study the graph without being too concerned about such cases, it becomes clearer that our hybrid method, in general, is capable of achieving high accuracy regardless of the content-based filtering algorithm (or the parameters) being used.

We also studied the configurations of the hybrid method with respect to the ratio of accuracy ($\Delta_{ACC} = ACC_{\text{hybrid}}/ACC_{\text{filteringonly}}$) and the ratio of traffic usage ($\Delta_{TU} = TU_{\text{hybrid}}/TU_{\text{filteringonly}}$).

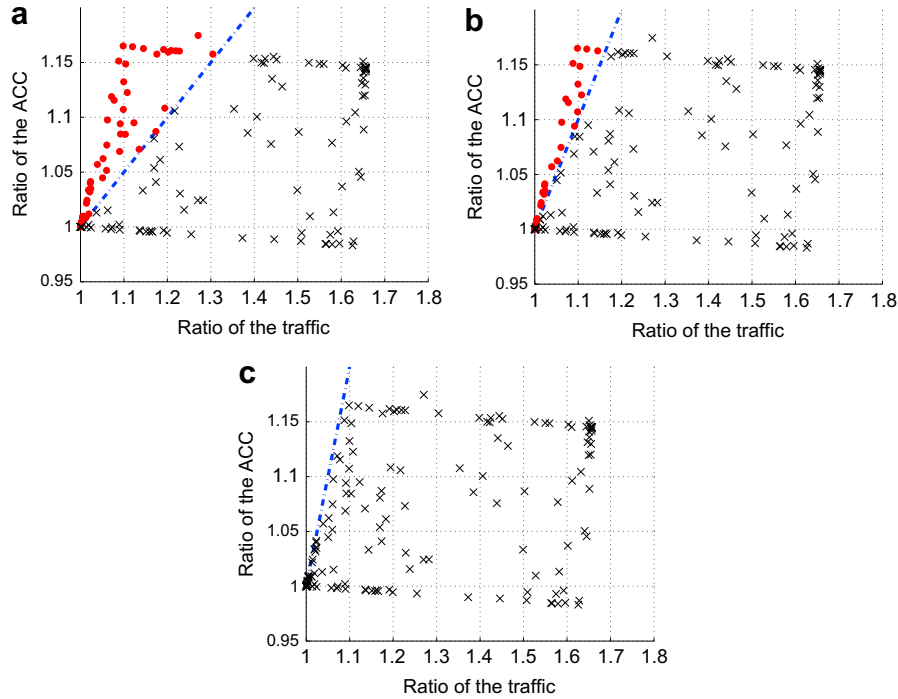


Fig. 10 – Comparison between the accuracy ratio and the traffic usage ratio for 200 samples — dotted blue lines stand for the borderline between the good samples and bad samples. (a) $\Delta_{ACC}:\Delta_{TU} = 1:2$. (b) $\Delta_{ACC}:\Delta_{TU} = 1:1$. (c) $\Delta_{ACC}:\Delta_{TU} = 2:1$.

5. Conclusion and future work

We proposed a hybrid spam filtering framework for mobile communication using a combination of *content-based filtering* and *challenge-response*. A message that falls into the uncertain region (after filtering), is further classified by sending a challenge (e.g. an image CAPTCHA) to the sender: a legitimate sender is likely to answer it correctly, whereas an automated spam program is not. Challenge-response protocols have been carefully designed with the necessary cryptographic features. We have also shown the trade-off between *accuracy* and *traffic*

usage in using our framework and compared it with the conventional content-based filtering method. Moreover, through a simulation of 200 randomly generated samples (each representing a unique set of content-based filtering parameters and threshold values) we showed that our hybrid approach, in general, achieves *high accuracy* regardless of the content-based filtering algorithm being applied. Although, when traffic usage becomes relatively more important than accuracy, the underlying filtering algorithms must be selected more carefully.

In this paper, a synthetic dataset, as opposed to a real dataset, has been used due to the following two reasons: first, we wanted to consider a wide range of application environments, each of which will require different level of accuracy and traffic usage (e.g. VoIP spam filters Croft and Olivier, 2005); and second, this protocol involves a great level of human interaction and developing such a prototype (in order to generate our own dataset) was outside the scope. As part of the future work, we could contact mobile operators and forums like OMTF to collect real data and verify the accuracy of our results.

Having the network operators charge for sending of SMS messages has been one of the big inhibitors to the growth of spam: even a cent per message might hugely alter the economics of a spammer. Assuming that a reasonable filtering method is in place, another hybrid potential is to force spammers to opt into a charging scheme where the cost of responding to a challenge is larger than sending an initial spam. For example, if it costs two cents to send a spam, then it would cost extra five cents to answer an image CAPTCHA. It is difficult to assess how effective a solution this might be, but future work may explore these economic measures in depth as a potential enhancement to the hybrid approach.

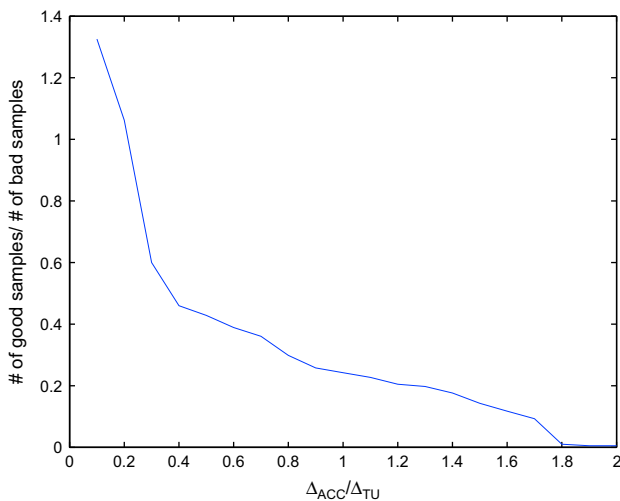


Fig. 11 – The ratio between the number of good samples and bad samples.

Acknowledgements

The authors would like to thank the anonymous referees for their careful attention and insightful comments.

**Appendix A.
Proof of protocol 1**

Protocol 1 has been proved by BAN logic (refer to the rules given by Burrows et al., 1989). In this protocol, M should be able to trust $H(S, R, P)$ returned from S, and know whether it is a legitimate sender or not. For formal verification, we derive an ideal protocol of protocol 1. The symbols N_s and N_m represent a sender's nonce and a recipient's nonce, respectively. Our goal is to show that this protocol satisfies the security property (G1).

[Ideal Protocol 1]

(M2) $M \rightarrow S : \{M \xleftrightarrow{K_{ms}} S\}_{K_c}, \{N_m, (S, R, P)\}_{K_{ms}}$

(M3) $S \rightarrow M : \{N_s, N_m, (S, R, P)\}_{K_{ms}}$

[Security Goal]

(G1) $M \equiv S \equiv (N_s, N_m, (S, R, P))$

Message 1 is ignored since it does not contribute much to achieving the goal; $\{N + 1\}$ is shown as N_s . The initial state assumptions have been derived: (A3) assumes that K_{ms} will be shared with a legitimate sender capable of

(A1) $M \equiv \#N_m$

(A2) $S \equiv \#N_s$

(A3) $M \equiv M \xleftrightarrow{K_{ms}} S$

(A4) $S \equiv M \Rightarrow M \xleftrightarrow{K_{ms}} S$

(A5) $M \equiv \xrightarrow{K_c} S$

(A6) $S \equiv \xrightarrow{K_c^{-1}} S$

interpreting K_c^{-1} . The proof is described as follows:

Sending message 2 leads to:

- (1) $M \mid \sim \{M \xleftrightarrow{K_{ms}} S\}_{K_c}$
- (2) $M \mid \sim \{N_m, (S, R, P)\}_{K_{ms}}$
- (3) $S \triangleleft \{M \xleftrightarrow{K_{ms}} S\}_{K_c}$
- (4) $S \triangleleft \{N_m, (S, R, P)\}_{K_{ms}}$

Sending message 3 leads to:

- (5) $S \mid \sim \{N_s, N_m, (S, R, P)\}_{K_{SR}}$
- (6) $M \triangleleft \{N_s, N_m, (S, R, P)\}_{K_{SR}}$

(7) is derived from (A3) and (6) by the message-meaning rule.

(7) $M \mid \equiv S \mid \sim (N_s, N_m, (S, R, P))$

$(N_s, N_m, (S, R, P))$ contains the nonce, N_s , and hence (8).

(8) $M \mid \equiv \#(N_s, N_m, (S, R, P))$

Finally (G1) is derived from (7) and (8) by the nonce-verification rule. In protocol 1, S cannot verify whether message 2 is from their contracted operator. Other protocols can be proved in a similar manner.

Appendix B.

Traffic amounts and accuracy of hybrid methods of 200 random samples (L: labels/classes)

Table 3.

Table 3 – Traffic amounts and accuracy of hybrid methods of 200 random samples (L: labels/classes).										
L	ID	μ_0	σ_0	μ_1	σ_1	h_1	h_2	TU	Ratio	ACC
1	11	0.6732	0.1385	0.6759	0.1187	0.3	0.6	3.517e+004	1.173	0.8921
	12	0.7174	0.05449	0.8433	0.1805	0.7333	0.8333	2.624e+004	1.135	0.8074
	13	0.7604	0.008962	0.8526	0.12	0.2667	0.4	3.005e+004	1.002	0.8565
	14	0.6399	0.09525	0.9397	0.1433	0.1333	0.7	3.316e+004	1.108	0.959
	15	0.637	0.1124	0.9365	0.04716	0.2333	0.5667	3.071e+004	1.024	0.8924
	16	0.5648	0.09243	0.8593	0.04778	0.5	0.7333	3.137e+004	1.077	0.9954
	17	0.739	0.1425	0.8176	0.06279	0.4	0.6333	3.056e+004	1.022	0.8878
	19	0.5905	0.07692	0.95	0.0202	0.1667	0.3667	3e+004	1	0.8572
	21	0.4844	0.06268	0.6896	0.07335	0	0.6667	3.914e+004	1.305	0.9921
	24	0.6414	0.02375	0.7796	0.04666	0.2667	0.5333	3e+004	1	0.857
2	8	0.5559	0.04695	0.6891	0.02269	0.5	0.8	4.883e+004	1.655	0.9816
	9	0.4918	0.0645	0.8896	0.0539	0.8667	0.9667	3.156e+004	1.564	0.7387
	18	0.6342	0.06492	0.7137	0.05991	0	0.8667	4.966e+004	1.655	0.9815
	23	0.4885	0.09164	0.7519	0.2259	0.3333	0.9	4.107e+004	1.444	0.9321
	28	0.8101	0.08229	0.8187	0.0469	0.4	1	4.97e+004	1.657	0.9814
	29	0.6651	0.09587	0.8033	0.08575	0.06667	0.8333	4.267e+004	1.422	0.9851
	31	0.5595	0.09035	0.8751	0.07934	0.7667	0.9333	3.515e+004	1.446	0.9038
	37	0.5251	0.1116	0.7205	0.1148	0.6667	0.8667	3.151e+004	1.528	0.7385
	41	0.6798	0.2155	0.7382	0.07893	0.3	0.8667	4.8e+004	1.612	0.9491
	51	0.5878	0.1235	0.7914	0.1444	0.6667	0.9333	3.48e+004	1.499	0.8222

Table 3 (continued)

L	ID	μ_0	σ_0	μ_1	σ_1	h_1	h_2	TU	Ratio	ACC
3	34	0.592	0.06966	0.6375	0.02727	0.8	0.9	5000	1	0.143
	58	0.5707	0.04015	0.709	0.04381	0.8667	0.9	5000	1	0.143
	66	0.3453	0.09073	0.8467	0.04666	0.9333	0.9667	5506	1.042	0.1542
	77	0.6668	0.06	0.7714	0.1038	0.9	0.9667	8408	1.213	0.2187
	86	0.7671	0.002867	0.7887	0.0686	0.9667	1	5000	1	0.143
	93	0.6524	0.01272	0.6852	0.1023	0.8667	1	5876	1.07	0.1622
	101	0.5471	0.03209	0.7869	0.05634	0.9667	1	5000	1	0.143
	106	0.7164	0.01455	0.8104	0.04486	0.9667	1	5000	1	0.143
	110	0.6261	0.06151	0.7295	0.1309	0.9	0.9333	7596	1.126	0.2119
	144	0.5295	0.06446	0.726	0.01037	0.8333	0.8667	5000	1	0.143
	4	80	0.6065	0.09277	0.7828	0.08489	0.8667	0.9333	1.129e+004	1.305
87		0.5061	0.07041	0.7851	0.01709	0.8	1	1.225e+004	1.353	0.302
124		0.5949	0.1187	0.6783	0.2052	0.7667	0.8667	1.711e+004	1.242	0.4799
130		0.4295	0.0666	0.7427	0.056	0.8	0.9667	1.071e+004	1.307	0.2682
131		0.5576	0.03068	0.6677	0.1524	0.8333	1	1.066e+004	1.305	0.2671
133		0.6727	0.006693	0.8018	0.0628	0.8333	0.9	1.725e+004	1.41	0.4273
139		0.6048	0.03089	0.7245	0.07272	0.8	0.9333	1.069e+004	1.306	0.2677
141		0.5513	0.07846	0.7591	0.1525	0.9	1	1.272e+004	1.365	0.3121
143		0.6858	0.1073	0.6896	0.2036	0.9	0.9667	1.033e+004	1.239	0.2742
152		0.5276	0.2158	0.8952	0.08286	0.9333	1	2.118e+004	1.508	0.4964

REFERENCES

- Androutsopoulos I, Koutsias J, Chandrinou K, Spyropoulos CD. An experimental comparison of naive Bayesian and keyword-based anti-spam filtering with personal e-mail messages. In: SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on research and development in information retrieval. New York, NY, USA: ACM; 2000. p. 160–7.
- Bratko A, Filipiè B, Cormack GV, Lynam TR, Zupan B. Spam filtering using statistical data compression models. *Journal of Machine Learning Research* 2006;7:2673–98.
- Burrows M, Abadi M, Needham R. A logic of authentication. *ACM Operating Systems Review* 1989;23(5):1–13.
- Chow R, Golle P, Jakobsson M, Wang L, Wang X. Making CAPTCHAs clickable. In: HotMobile '08: Proceedings of the 9th workshop on mobile computing systems and applications. New York, NY, USA: ACM; 2008. p. 91–4.
- Cormack GV, Hidalgo JMG, Sanz EP. Spam filtering for short messages. In: Proceedings of the 16th ACM conference on information and knowledge management; 2007. p. 313–20.
- Croft NJ, Olivier MS. A model for spam prevention in IP telephony networks using anonymous verifying authorities. In: ISSA, new knowledge today conference; 2005.
- Deng W, Peng H. Research on a naive Bayesian based short message filtering system. In: Machine learning and cybernetics, 2006 international conference on Aug. 2006. p. 1233–7.
- Dwork C, Goldberg A, Naor M. On memory-bound functions for fighting spam. In: Proceedings of the 23rd annual international cryptology conference (CRYPTO 2003); August 2003.
- Enck W, Traynor P, McDaniel P, Porta T. Exploiting open functionality in SMS-capable cellular networks. In: CCS, Nov. 2005.
- Golbeck J, Hendler J. Reputation network analysis for email filtering. In: Proceedings of the conference on email and anti-spam (CEAS); 2004.
- Hall RJ. How to avoid unwanted email. *Communications of the ACM* March 1998.
- Hartigan JA. Clustering algorithms. New York: John Wiley and Sons; 1975.
- He P, Sun Y, Zheng W, Wen X. Filtering short message spam of group sending using CAPTCHA. In: Workshop on knowledge discovery and data mining; 2008. p. 558–61.
- Healy M, Delany S, Zamolotskikh A. An assessment of case-based reasoning for short text message classification. In: Proceedings of 16th Irish conference on artificial intelligence and cognitive science; 2005. p. 257–66.
- Hidalgo JMG, Bringas GC, Sanz EP, Garc FC. Content based SMS spam filtering. In: Proceedings of the 2006 ACM symposium on document engineering. Amsterdam, The Netherlands: ACM Press; October 2006. p. 10–3.
- Leveraging the CAPTCHA Problem; 2005.
- Metsis Vangelis, Androutsopoulos Ion, Paliouras Georgios. Spam filtering with naive Bayes – which naive Bayes? In: Third conference on email and anti-spam (CEAS); 2006.
- Prieto AG, Cosenza R, Stadler R. Policy-based congestion management for an SMS gateway. In: Proceedings of the fifth IEEE international workshop; 2004.
- Rogers D. Mobile handset security: securing open devices and enabling trust. OMT Limited White Paper 2007.
- Roman R, Zhou J, Lopez J. An anti-spam scheme using pre-challenges. *Computer Communications* 2006;29(15):2739–49.
- Shirali-Shahreza S, Movaghar A. An anti-SMS-spam using CAPTCHA. In: CCCM '08: Proceedings of the 2008 ISECS international colloquium on computing, communication, control, and management. Washington, DC, USA: IEEE Computer Society; 2008. p. 318–21.
- von Ahn L, Maurer B, Mcmillen C, Abraham D, Blum M. RECAPTCHA: Human-based character recognition via web security measures. *Science* August 2008.
- Yan J, El Ahmad AS. Usability of CAPTCHAs or usability issues in CAPTCHA design. In: SOUPS '08: Proceedings of the 4th symposium on usable privacy and security. New York, NY, USA: ACM; 2008. p. 44–52.
- Ji Won Yoon.** He received the B.Sc. degree in information engineering at the SungKyunKwan University, Korea. He obtained the M.Sc. degree in School of informatics at the University of

Edinburgh UK in 2004 and the Ph.D. degree in signal processing group at the University of Cambridge UK in 2008 respectively. In 2008, he moved to department of Engineering science, the University of Oxford, UK to do postdoctoral research.

He is currently a Research Fellow with Statistics department, Trinity College Dublin, Ireland. His research interests include Bayesian statistics, Machine Learning, data mining, Network Security and Biomedical engineering. He has worked on applications in brain signals, cosmology, biophysics and multimedia.

Hyoungshick Kim. He received the B.Sc. degree in information engineering at the SungKyunKwan University, Korea. He obtained the M.Sc. degree in department of computer science, KAIST, Korea in 2001. He previously worked for Samsung Electronics as a senior

engineer from May 2004 to September 2009. He also served a member of DLNA and Coral standardization for DRM interoperability in home networks.

He is currently studying in the Computer Laboratory at the University of Cambridge as a PhD student. His research interest is focused on security and privacy in complex networks and distributed systems.

Jun Ho Huh. He holds Software Engineering and International Business degrees from Auckland University. He is currently a DPhil student at the Oxford University Computing Laboratory. His research interests include trusted virtualization, trustworthy audit and logging, and security in distributed systems.