Resolving FP-TP Conflict in Digest-Based Collaborative Spam Detection by Use of Negative Selection Algorithm

Slavisa Sarafijanovic
EPFL, Switzerland
slavisa.sarafijanovic@epfl.ch

Sabrina Perez
EPFL, Switzerland
sabrina.perez@epfl.ch

Jean-Yves Le Boudec
EPFL, Switzerland
jean-yves.leboudec@epfl.ch

ABSTRACT

A well-known approach for collaborative spam filtering is to determine which emails belong to the same bulk, e.g. by exploiting their content similarity. This allows, after observing an initial portion of a bulk, for the bulkiness scores to be assigned to the remaining emails from the same bulk. This also allows the individual evidence of spamminess to be joined, if such evidence is generated by collaborating filters or users for some of the emails from an initial portion of the bulk. Usually a database of previously observed emails or email digests is formed and queried upon receiving new emails.

Previous evaluations [2, 10] of the approach based on the email digests that preserve email content similarity indicate and partially demonstrate that there are ways to make the approach robust to increased obfuscation efforts by spammers. However, for the settings of the parameters that provide good matching between the emails from the same bulk, the unwanted random matching between ham emails and unrelated ham and spam emails stays rather high. This directly translates into a need for use of higher bulkiness thresholds in order to ensure low false positive (FP) detection of ham, which implies that larger initial parts of spam bulks will not be filtered, i.e. true positive (TP) detection will not be very high (FP-TP conflict).

In this paper we demonstrate how, by use of the negative selection algorithm, the unwanted random matching between unrelated emails may be decreased at least by an order of magnitude, while preserving the same good matching between the emails from the same bulk. We also show how this translates into an order of magnitude (at least) of less undetected bulky spam emails, under the same ham miss-detection requirements.

Keywords

Email, spam, open digest, similarity hashing, data representation, collaborative, detection, filtering, obfuscation, robustness, negative selection algorithm.

In the Proceedings of CEAS 2008, the Fifth Conference on Email and Anti-Spam, Aug 21-22, 2008, Mountain View, California, USA.

1. INTRODUCTION

1.1 Importance of Digest-Based Collaborative Spam Detection

There are two big groups of techniques for collaborative communication filtering in the Internet (including the email filtering): those based on the source of the communication and those based on the content of the communication. The techniques from the first group try to determine how good or reputed the source of a communication is. Based on this evaluation the communication is either protected (white listed), or blocked, or given a score that is combined with other antispam techniques for the final filtering decision. The reputation is usually determined for sender IDs, source IP addresses or Internet domains. Concrete examples are PGP-based trust and authentication for email addresses [5], Spamhouse's real-time black lists of IP addresses from which large amount of spam is sent [11], Gmail's reputation of Internet domains from which emails are sent [12].

The techniques from the first group are however able to filter only the communication from the sources already learned to be good or bad, though in practice it is often needed to accept and filter new communications from the users or sources for which such a reputation is not yet built. Regarding email, the white and black lists are usually applied first as they are both fast and allow pre-filtering the email stream already at the connection level, which is good for lowering the usage of the communication resources. However, due to the use of dynamic domains and botnets by spammers, significant amount of spam is not blocked by black lists. Use of additional spam filtering techniques, such are those based on email content, is still needed.

Collaborative detection of bad content seems to be especially appropriate for detecting and blocking bulk spam (most of spam is sent in bulk). It allows for a relatively early detection of new bulks that contain not previously observed spam message (phrase).

1.2 FP-TP Conflict And Need To Lessen It

In content-based collaborative spam detection, usually a database of previously observed emails or email digests is formed, and queried upon receiving new emails. This allows, after observing an initial portion of a bulk, for the bulkiness scores to be assigned to the remaining emails from the same bulk. Spam will be distinguished by having "many (more then a threshold) similar emails have been observed" score, while a ham query should score "less then the (same) threshold similar emails has been observed". This also allows the individual evidence of spamminess to be joined, if such evidence is generated by collaborating filters or users for some of the emails from an initial portion of the bulk. A good source of the evidence of spamminess, which is increasingly used in practice, are the emails tagged as spam by those users that have and use a "delete-as-spam" button in their email-reading program. Automated and probabilistic tagging is also possible, e.g. by use of Bayesian filters' scores, or by use of "honey pot" email accounts that are not associated to real users but only serve to attract unsolicited bulk emails.

It should be mentioned that if the collaborative spam detection is based purely on the evaluation of bulkiness, each recipient should be equipped with a white lists of all the bulky sources from which she or he wants to receive emails. If trustworthy spamminess reports are used, the scheme may be usable even without the white lists.

Whether some of the previously observed emails are tagged as spam or not, for the collaborative detection to work it is necessary to have a good technique for determining similarity between the emails and find which emails (probably) belong to the same bulk, i.e. spam emails from the same bulk should "match" each other with a high probability, and ham emails should with a high probability not match unrelated ham and spam emails.

It has been indicated and partially demonstrated [2, 10] that collaborative spam bulk detection by use of similarity digest may be efficient even under strong obfuscation by spammer, i.e. when the spammer modifies the different copies from a bulk in order to hide their mutual similarity from the automated tools. However, the same evaluations suggest that for the settings of the parameters that provide good matching between the emails from the same bulk, the unwanted random matching between ham emails and unrelated ham and spam emails stays pretty high. This directly translates into a need for use of higher bulkiness thresholds in order to ensure low false positive (FP) detection of ham, which implies that larger initial parts of spam bulks will not be filtered, i.e. true positive (TP) detection will not be very high (FP-TP conflict).

The above facts suggest need for use of additional algorithms (as opposed to simple counting of similar digests in the database), in order to lessen the FP-TP conflict and make the technique more useful.

1.3 Nilsimsa Hashing For Determining Similarity Between Emails

Single digest per email. The open-digest technique from the OD-paper represents an email by a 256-bits digest. The transformation is performed using Nilsimsa hashing [4]. This is a locally sensitive hash function, in sense that small changes in the original document may impact only few bits of the digest. That means that similar documents will have similar digests, in sense of a small Hamming distance between them. With the standard hash functions small changes in the original document usually result in a digest that is completely different from the digest of the original document.

Nilsimsa Hashing. OD-paper gives a detailed description of the Nilsimsa hashing. In summary, a short sliding window is applied through the email. For each position of the window, the trigrams from the window are identified that consist of the letters from the predefined window positions (that are close to each other, but not only consecutive-letters trigrams are used). The collected trigrams are transformed, using a standard hash, to the positions between 1 and 256, and the accumulators at the corresponding positions are incremented. Finally, the accumulators are compared to the mean or to the median of all the accumulators, and the bits of the digest are set to 0 or 1, depending on whether the corresponding accumulators are bellow or above the mean (or median).

Such digest are often called "open digests" because: a) the digests computation method is assumed to be publicly known; b) the used similarity hashing hides original email text, so the privacy of the content is preserved even if the digests are openly exchanged for collaborative filtering.

Nilsimsa Compare Value (NCV) between two digests is defined to be equal to the number of the equal bits at the same positions in the two digests, minus 128 (for the digests of 256 bits). We use NCV as the measure of the similarity of the two emails from which the two digests are produced[1]. The higher NCV indicates the higher similarity of the texts from which the digest are computed. NCV of two random unrelated text-strings is random and centered around 0. Distribution of NCV for two readable (meaningful) but unrelated texts from the same language is slightly shifted to the right (in our examples centered around 30). If the texts are related (contain similar or identical parts) NCV takes values considerably higher then 30, and if the texts are completely identical NCV is equal to 128.

Digests produced from randomized-position fixed-length samples. In our recent paper [10] paper we demonstrate that using multiple digests produced from the randomized-position and fixed-length samples is more resistant to obfuscation then use of the above explained single hash, regarding spam bulk detection under an interesting and practical obfuscation model[2]. In the same paper we explain that such digests should also be beneficial under other obfuscation models, though that is not demonstrated experimentally.

Email-to-email NCV. In order to evaluate similarity between two emails when multiple digests per email are used, we define in [10] the NCV between two emails to be the maximum NCV over all the pairs of the digests between the two compared emails. Such defined email-to-email NCV shows how similar are the most similar parts in the two emails. We use the same email-to-email NCV similarity measure in this paper.

1.4 FP-TP Conflict And Existing Techniques For Content-Based Bulk Spam Detection

Here we list the techniques that are known to us to use similarity digests and that are evaluated in the scientific literature. Simple counting of the recently observed digests in a common queried database using single hash per email is evaluated by Damiani et al. in the well known and often cited OD-paper [2], and recently revised by us in [10]. In [10] we also evaluate the same case but with randomized-position and fixed-length digests (multiple per email).

Zhou et al. [13] design and evaluate a peer-to-peer digest-based system for collaborative spam detection, which also uses randomized-position and fixed-length digests. However, it uses exact instead similarity matching when comparing

[1] NCV is first defined and used in open-digest paper [2].

[2] We considered addition of random text to the original spam message, the case that is often observed in practice.

the digests (as required for hash based routing used by their system to work).

We first introduce the idea of using multiple email digests and negative selection together in [8]. The system that we design and evaluate in [9] produces multiple digests per email, from the strings of fixed length, sampled at random email positions, and it uses similarity matching. Additionally, it uses artificial immune system algorithms to process the digests before and after exchanging them with other collaborating systems, in order to control which digests will be activated and used for filtering of the incoming emails.

The previously mentioned FP-TP conflict due to the random matching between unrelated emails remains in effect in all these approaches. Actually, in [9] we use the negative selection algorithm to lessen the effect of this conflict, but as the algorithm is used in interplay with other artificial immune system algorithms and the factorial analysis is missing, the effect of the negative selection algorithm itself is not clarified.

1.5 Our Approach For Lessening FP-TP Conflict And Contributions of This Paper

In this paper we follow our suggestion from [10] and evaluate the impact of the negative selection on lessening the FP-TP conflict (FP-TP conflict is explained in Section 1.2).

Use of negative selection algorithm in this case[3] simply means that the digests from the newly received emails are first compared to a database of the digests created from a representative set of good emails (so called SELF database), and those that match any of the SELF database digests are deleted, and only the remaining digests are used for querying the collaborative database of previously observed digests (Figure 1).

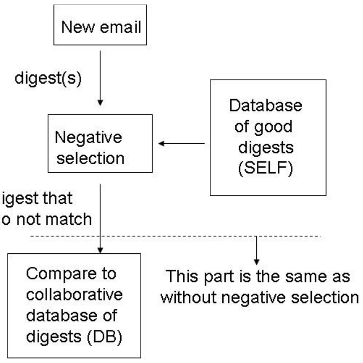

New email
digest(s)
Negative selection
Database of good digests (SELF)
digest that do not match
Compare to collaborative database of digests (DB)
This part is the same as without negative selection

Figure 1: Adding negative selection to the detection process

In our experiments we form the SELF set by taking randomly a number of good emails from a part of the used ham corpus (the other two ham corpus parts are used for simulating incoming ham that is to be filtered and for simulating the digests that are created from previously observed emails and still present in the collaborative-detection database). In practice SELF could be automatically updated (e.g. the emails we send and reply to could be used to update the SELF set).

We consider the case of collaborative spam bulk detection that uses multiple digests produced from the strings of fixed length sampled at randomized positions within email, because this is shown [10] to be more resistant to increased spam obfuscation then using single digest per email (for the considered spammer model). We consider the same spammer model as the one used in [10], i.e. addition of random text to the original spam message, but we also explain the expected results for other interesting spammer models.

We perform the same experiment as the one used in [10], but both with and without use of negative selection. We evaluate spam bulk detection by estimating the probability for the emails from the same bulk to match each other, and we evaluate miss-detection of good emails by evaluating the probability for ham emails to match unrelated ham and spam emails.

Examples of digests that could cause unwanted ham to "unrelated" emails matching, and that the negative selection is expected to eliminate, are the digests produced from sender's email-client information and used email-formatting information, which are often present in email headers, or the digest produced from the interesting textual content that is being epidemically forwarded to the friends (ham examples may also be dynamically collected, e.g. from authenticated and reputable users). There are also some standard phrases that are often used in the good emails (introductory parts and salutations).

We demonstrate that by use of the negative selection algorithm the unwanted random matching between unrelated emails may by decreased at least by an order of magnitude, while preserving the same good matching between the emails from the same bulk. We also show how this translates into an order of magnitude (at least) less undetected bulky spam emails, under the same ham miss-detection requirements.

1.6 How Our Approach is Different from Cloudmark's "negative assertion"

Prakash and O'Donnell [7] describe a digest-based reputation system called CNC (Cloudmark Network Classifier). CNC assumes that some users protected by the system report about the received emails whether they are spam or not. A reputation of the reporters is maintained and used to aggregate the reports on the same signatures (the system uses exact matching between the similarity-preserving signatures). The reporting of non-spam messages in their system is called "negative assertion". The negative assertion is used with the purpose of avoiding false detections of bulky good email.

The main difference between CNC and our our approach is in the fact that CNC does not process the signatures locally before submitting them to the central database and our system does. While the "negative assertion" declares all the signatures from one email as spammy or normal, our approach is more precise in that it selects (by use of negative selection) which signatures should be used further, and additionally processes them (creation and local processing of antibodies) before deciding to exchange them for collaborative filtering. This difference is important because there are some email parts that may often be similar in both good and bad emails (as discussed in the previous section), that spammers may even intentionally add to the spam emails in order to pollute the system. Our system is able to remove such email parts from further processing, and thus isolate and more precisely process the specific email content.

[3] Negative Selection is a well known AIS algorithm, e.g. see [3], [9]; AIS stands for Artificial Immune Systems, which are the systems built using the concepts and algorithms inspired by the models of the human immune system.

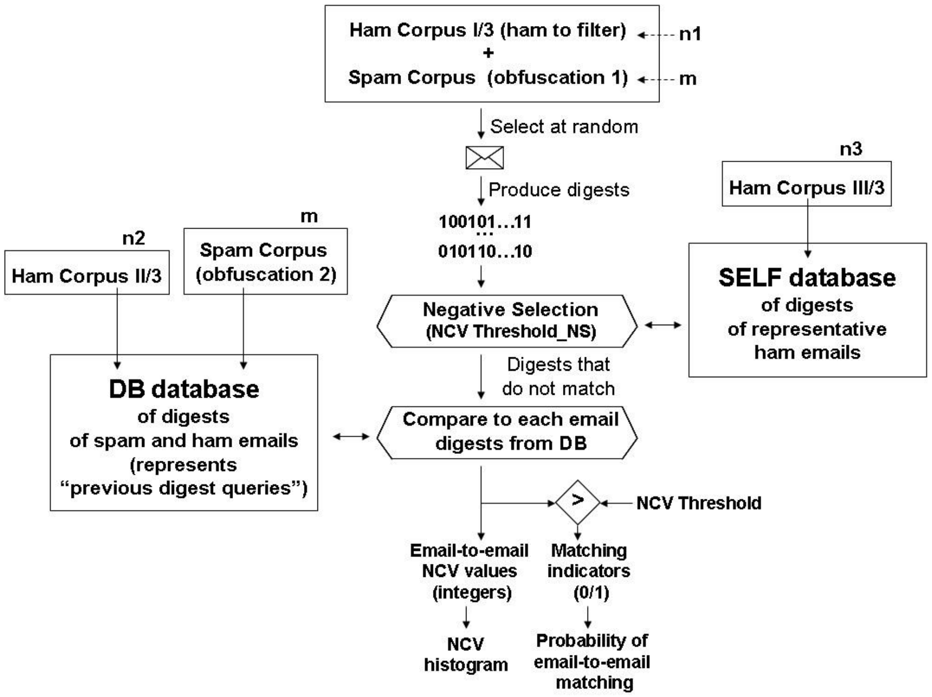


Figure 2: Experiment without/with negative selection. **The experiment determines the probabilities of matching between ham or spam emails and the emails from the collaborative-detection database DB. If the negative selection step is used, some of the digests from some emails will be deleted before comparing the email to the database. We also look at the NCV histograms in order to better understand what happens without/with negative selection. The ni ($i = 1, 2, 3$) and m are the number of emails used in the comparisons (in this experiment these values are set to 20).**

2. EXPERIMENTAL EVALUATION OF NEGATIVE SELECTION

In this section we give additional details of the approach and its experimental evaluation introduced in Section 1.5. The code used for performing the experiments is available on the web [1].

2.1 Experiments Setup

The performed experiments (with and without use of negative selection) are illustrated in Figure 2.

The experiments determine the probabilities of matching between ham or spam emails and the emails from the collaborative-detection database DB. More precisely, each of n1 ham emails (that simulate newly received emails) is compared to each of n2 ham and m spam emails from the collaborative-detection database DB, in order to estimate the probability of matching between newly received ham emails and unrelated ham and spam email digests previously observed (within a time window) by the collaborative-detection database DB (we assume the DB does not a priori know which digests are spam and which are not, and it stores all of them). Also, each of m spam emails is compared to one obfuscated copy in DB database that originates from the same original spam message (two obfuscated copies from the same bulk), in order to estimate the probability of spam emails to match other spam emails from the same bulk. We used $ni, m = 20$. For all estimated probabilities we also compute confidence intervals. In Section 2.3 we show how these estimated email-to-email matching probabilities may be translated into ham miss-detection and spam bulk detection ratios.

The emails are selected randomly from the corresponding databases. Spam emails are optionally obfuscated (as specified in Section 1.5). The comparison is on the level of similarity digests and is expressed in form of email-to-email NCV (this similarity measure is defined in Section 1.3). We recall here that we generate multiple digests per email, the number of which is not fixed due to the randomized sampling of the email text.

Optionally, negative selection is applied to remove some of the email digests before comparing that email to the emails from the database DB. The negative selection is aimed at removing those digests that show tendency to match the digests from good emails, i.e. they are removed if they match any of the SELF-database digests (SELF-database digests are produced from emails known to be ham). If all digests would happen to be removed, the message could not be judged by this method. This could happen for messages with a very short content part, or for image spam. This is however expected to be unlikely to happen for bulky textual spam that we consider in our experiments.

In all experiments we use the detection NCV threshold 90, and when the negative selection is applied we use the negative selection NCV threshold 50. We choose these values as they provide representative results, but we do not optimize them.

Evaluation Metrics. In addition to estimating *email-to-email matching probabilities*, we also show the *histogram of email-to-email NCVs* (in the case of spam-detection evaluation, we account only for comparisons against the emails from the same bulk - otherwise bulk-matching results would be masked by unrelated-emails-matching results). Histograms are useful for understanding what exactly happens with the digests and for explaining the results.

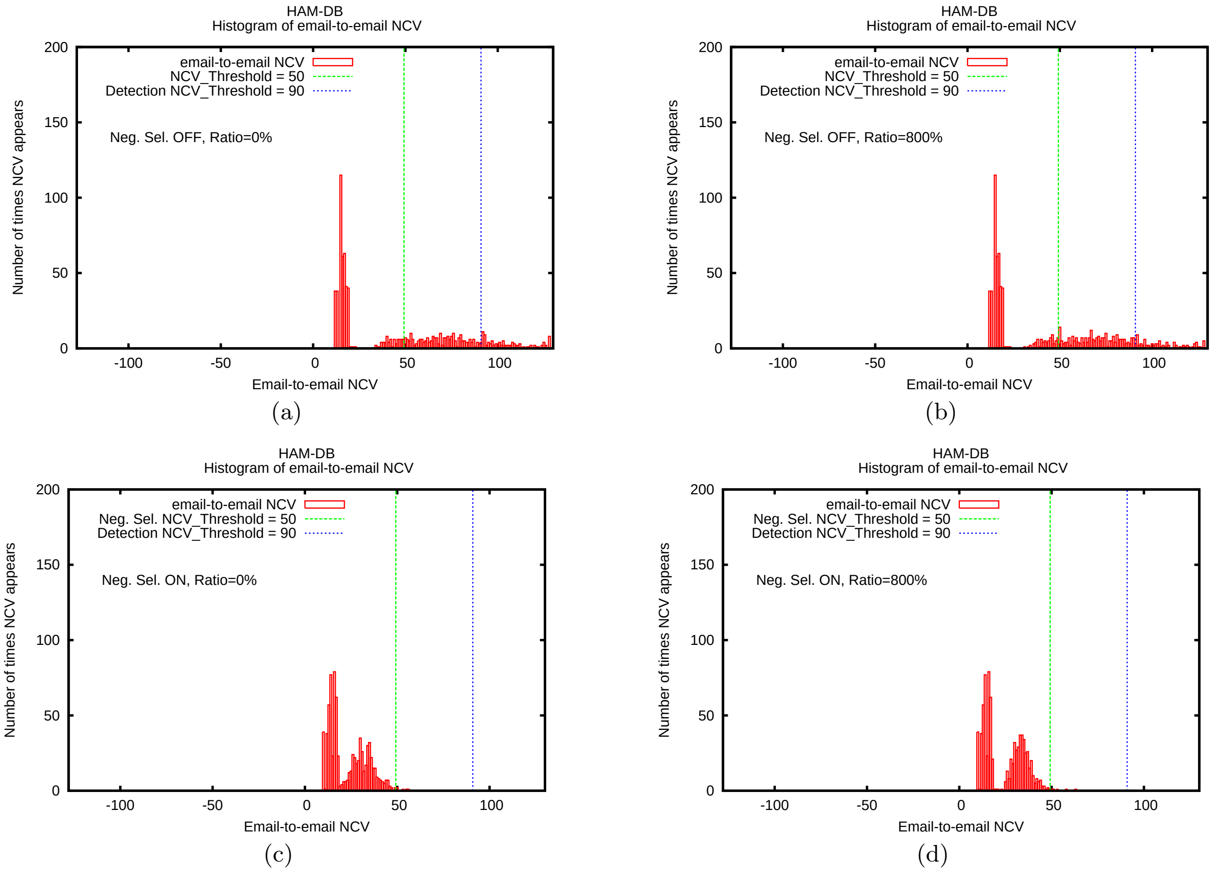


Figure 3: Impact of Negative Selection on miss-detection of ham: NCV histograms of (unwanted) random matching of ham emails to emails in the collaborative-detection database DB.

2.2 Results Discussion

2.2.1 *Impact of Negative Selection on Ham Miss-Detection*

From Figures 3(a) and 3(b) we can see that, when the negative selection is not used, the histogram of the NCV values between ham emails and unrelated ham and spam emails is very wide and have a tail in the right part of the figures, which explains the high values of the probability of (unwanted) matching between the ham emails and unrelated ham and spam emails (dashed green line of the Figure 4). From Figures 3(a), 3(b) and 4 we can also see that the obfuscation doesn't impact the the probability of (unwanted) matching between the ham emails and unrelated ham and spam emails.

From Figures 3(c) and 3(d) we can see that, upon applying the negative selection, most of the ham digests that cause unwanted matching between the ham emails and unrelated ham and spam emails are deleted (remaining NCV similarity between unrelated emails is bellow the detection threshold).

Actually we observe no matching between ham emails and unrelated ham and spam emails from the database DB (solid red line of the Figure 4) overlaps with x axes). However, when one type of output is not observed in repeated binary experiment (in our case matching of unrelated emails upon applying the negative selection), that doesn't mean that the probability of that event is zero. It might be very low to be detected with a relatively small number of experiment repetitions. The upper value of the 95% confidence interval to which the real value of the probability belongs may be computed using the following formula [6]:

$$CI_{upper} = 1 - (\alpha/2)^{(1/n)} \qquad [1]$$

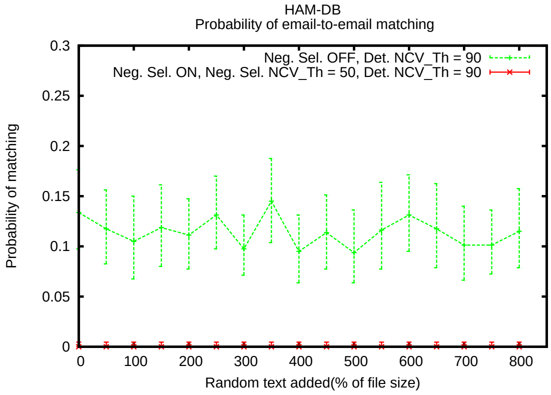
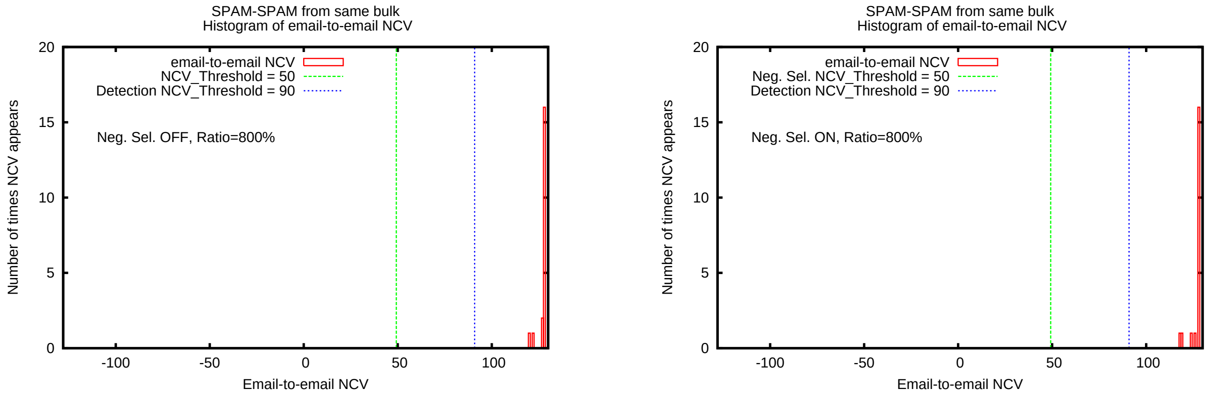


Figure 4: Impact of Negative Selection on miss-detection of ham: The probability of (unwanted) random matching of ham emails to emails in the collaborative-detection database DB drops significantly when the negative selection is applied.



(a) t

(b)

Figure 5: Impact of negative selection on detection of spam bulk: NCV histograms of matching spam emails to the emails from the same bulk.

In our case $\alpha = 0.05$ (as we want to compute 95% confidence interval), and $n = 800$ (as we compare 20 ham emails to 40 emails from the database DB). We obtain $CI_{upper} = 0.0046$. The lower limit of the confidence interval is equal to 0. In order to more precisely estimate the real value of the probability, larger number of comparisons should be performed. However the determined CI already shows that negative selection causes a large decrease of the probability for ham emails to match unrelated ham and spam emails from DB.

2.2.2 *Impact of Negative Selection on Spam Bulk Detection*

From Figure 5(a) we see that even under strong obfuscation (ratio=800%) the matching between the emails from the same bulk is very strong (high email-to-email NCV values), i.e. the considered digests are very resistant to the increased obfuscation (this is the finding of our previous paper [10] that we repeat here for the reason of comparison of the results without and with negative selection).

From Figure 5(b) we see that turning on the negative selection does not have visible effects on the matching between the spam emails from the same bulk.

For the used detection NCV thereshold 90, in cases with and without negative selection, all observed email-to-email NCV values are above the threshold and the observed matching ratio is equal to 1 (so we do not plot this value).

2.3 Transforming (estimated) Email-to-email Matching Probabilities Into FP and TP

To determine whether an email is spam or not based on its observed bulkiness, the number of the emails that it matched in the database DB should be compared to a bulkiness threshold. If this number is above the threshold, that is indication the email is probably spam, else it is probably ham.

If the number of the emails in the database DB is equal to N, and the probability of matching between a ham and un unrelated ham or spam email is equal to p, and assuming the result of email comparisons between a ham email and the emails from the database are approximately independent and identically distributed binary variables, the probability of the query score for a ham email to be above the bulkiness threshold th, which is at the same time the probability of miss-detection of ham emails, is equal to:

$$FP = 1 - BinoCdf(th, N, p), \qquad [2]$$

where BinoCdf is the well known binomial cumulative distribution function.

For example if we consider the case of N=100000, for the two values of p that we obtained in the performed experiments (see Figure 4): p1=0.1 without negative selection and p2=0.0046 with negative selection, we obtain the probability of ham miss-detection in function of the bulkiness threshold as shown in the Figure 6 (the curves are computed and shown only for the probabilities that are not bellow the precision of the computer).

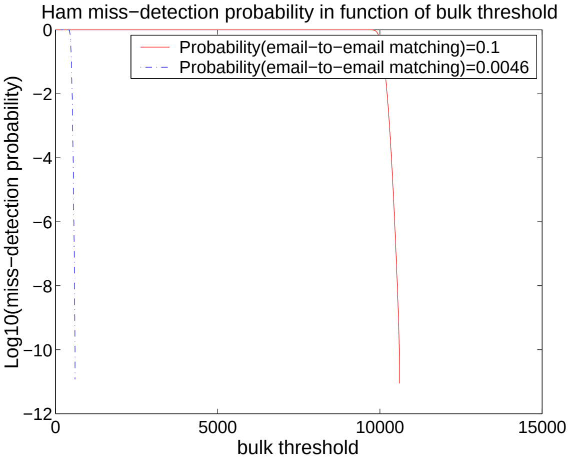


Figure 6: Determining bulkiness threshold that satisfies a low ham miss-detection requirement: When the negative selection is applied (the dashed and doted line), a low ham miss-detection probability can be achieved with a significantly smaller value of the bulkiness threshold. This means that a much smaller initial part of a spam bulk will make it into inboxes (as compared to the case without negative selection) before enough evidence is collected for a safe detection of the remaining messages from the same bulk.

We see that, in order to achieve very small ham miss-detection probability, negative selection allows use of about 20 times smaller bulkiness threshold. This directly translates into approximately 20 times smaller initial portion of the bulk that will not be filtered until enough evidence is collected. In the case with negative selection the real probability of matching between ham emails and unrelated ham or spam emails might be even smaller the used value p2=0.0046 (we used the upper limit of the confidence interval), the gain might be even bigger.

2.4 Expected Behavior under Other Spammer Models

It would be interesting to evaluate the negative selection under different spammer models, especially under those that decrease the probability p of matching between emails from the same bulk (obfuscation usually do not impact the probability of matching between ham and unrelated ham and spam emails). In that case the initial part of the bulk that would not be filtered (using bulkiness threshold that provides small ham miss-detection) would be approximately increased by factor $1/p$.

2.5 Expected Behavior within Other Antispam Schemes

It seems that the negative selection mechanism can easily be added (as a module) to other digest-based collaborative detection schemes, including peer-to-peer schemes and the schemes that use various additional inputs (e.g. user feedback) and algorithms as opposed to a simple counting of bulkiness. As the negative selection mechanism achieves its effect through the decrease of unwanted matching between the digests from unrelated emails, its benefit should hold when it is used within any digest-based collaborative detection scheme.

3. CONCLUSION

In this paper we demonstrate how by use of the negative selection algorithm the unwanted random matching between unrelated emails may by decreased at least by an order of magnitude, while preserving the same good matching between the emails from the same bulk. We also show how this translates into an order of magnitude (at least) less undetected bulky spam emails, under the same ham miss-detection requirements. Moreover, it seems that a use of multiple digests per email and the negative selection mechanism could be beneficial for any digest-based collaborative spam detection scheme.

4. REFERENCES

[1] Source code of the experiments from this paper: http://icawww.epfl.ch/ssarafij/negative-selection-of-digests/ , April 2008.

[2] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, and P. Samarati. An open digest-based technique for spam detection. In *Proceedings of The 2004 International Workshop on Security in Parallel and Distributed Systems*, San Francisco, CA, USA, September 2004.

[3] J. Kim and P. J. Bentley. An evaluation of negative selection in an artificial immune system for network intrusion detection. In L. Spector, E. D. Goodman, A. Wu, W. B. Langdon, H.-M. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. H. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, pages 1330–1337, San Francisco, California, USA, 7-11 2001. Morgan Kaufmann.

[4] Nilsimsa. http://lexx.shinn.net/cmeclax/nilsimsa.html, Sep 2006.

[5] OpenPGP. IETF RFC 4880, http://tools.ietf.org/html/rfc4880.

[6] A. M. Pires. Confidence intervals for a binomial proportion: comparison of methods and software evaluation. In *In Klinke, S., Ahrend, P. and Richter L. (editors), Proceedings of the Conference CompStat 2002*, August 24-28, 2002.

[7] V. V. Prakash and A. O'Donnell. Fighting Spam with Reputation Systems. In ACM Queue 3(9):36-41, 2005.

[8] S. Sarafijanovic and J.-Y. L. Boudec. Method to Filter Electronic Messages in A Message Processing System, US Patent No. 2008/0059590 A1, filed September 5, 2006, published March 3, 2008.

[9] S. Sarafijanovic and J.-Y. Le Boudec. Artificial immune system for collaborative spam filtering. In *Proceedings of NICSO 2007, The Second Workshop on Nature Inspired Cooperative Strategies for Optimization, Acireale, Italy, November 8-10, 2007.*

[10] S. Sarafijanovic, S. Perez, and J.-Y. Le Boudec. Improving digest-based collaborative spam detection. In *Proceedings of The 2008 MIT Spam Conference*, Cambridge, Massachusetts, USA, March 27-28, 2008.

[11] Spamhaus. http://www.spamhaus.org/ , April 2008.

[12] B. Taylor. Sender Reputation in a Large Webmail Service. In Proceedings of CEAS 2006, the Third Conference on Email and Anti-Spam, July 27-28, 2006, Mountain View, Califorina, USA.

[13] F. Zhou, L. Zhuang, B. Zhao, L. Huang, A. Joseph, and J. Kubiatowicz. Approximate object location and spam filtering on peer-to-peer systems. In *Proceedings of ACM/IFIP/Usenix Int'l Middleware Conf., LNCS 2672, pp. 1-20.*