A Two-Tiered User Feedback-based Approach for Spam Detection

Malik A. Feroze, Zubair A. Baig and Michael N. Johnstone

Security Research Institute &
School of Computer and Security Science
Edith Cowan University
Perth, Australia
Email: mferoze@our.ecu.edu.au, {z.baig, m.johnstone}@ecu.edu.au

***Abstract*—The current practice for spam detection works through binary classification of a message as either spam or ham. We propose a novel technique based on solicitation of user feedback in the spam classification process. The spam classifier proposed is semi-automated in nature, and is trained dynamically to include words and word-variants into the spam dictionary. Thresholds are defined to ascertain that spam and ham messages are accurately classified with highest probability. In addition, a set of messages that do not fall into the above two categories are tagged as grey messages. These messages are reclassified as ham or spam based on user feedback. Results obtained through experiments proved the superiority of the two-tier spam classifier over the single-tier spam classifier.**

***Index Terms*—Spam detection; User Feedback; Classification.**

I. Introduction

Spam is defined as unsolicited email intended for delivery to a large number of recipients. The classification of emails into spam and ham has remained a challenging task. Whilst common labels and frequently-occurring spam words can be identified with ease, the growing number of spam messages with well-crafted subject lines and message payloads, conveniently circumvent current spam classifiers. It is estimated that nearly 70% of global emails are spam, which equates to approximately 14.5 billion spam emails a day [1]. The annual cost due to loss in productivity through spam is estimated to be around $20 billion. This is because a percentage of an employee's time is spent browsing and deleting individual spam messages during a given day at work. Spam is not limited to menacing messages that originate from unknown sources. Rather, recent spam messages have been observed to be originating from legitimate domains such as those belonging to banks and other financial institutions [2]. Trojans operating clandestinely from the back-end servers of established businesses generate spam messages, with sensitive customer details, such as user names and phone numbers, listed in the message text. Given the cost of dealing with spam, loss of productivity and potential loss of confidentiality, the issue of spam identification is critical in contemporary times more than ever.

On a typical web-based form, input is validated via a regular expression parser or whitelisting to avoid attacks such as SQL injection. We propose a two-tier mechanism for identifying spam and improving the accuracy of existing spam classifiers. The proposed scheme solicits user feedback to train a spam classifier to accurately classify those messages that had initially been classified as belonging to neither the spam nor the ham message category. User intervention in training a spam classifier can prove to be successful provided that the usability of the proposed solution is not overly affected by imposition of added work onto an end-user. The scheme operates through definition of system parameters and classification policy, that helps categorize incoming messages into the grey list. Tier-2 of the scheme solicits user feedback and incorporates the outcome of its analysis into the decision-making engine.

One of the purposes of the proposed scheme is to pre-validate input prior to allowing users access to an internal system, thus providing a higher level of security through an additional layer of authentication. This mechanism uses a rules-based algorithm to determine if input either is valid, or should be blacklisted or even grey-listed. In terms of authentication systems, the first case is where legitimate credentials are presented and accepted. The second case is where a fraudulent (adversary) user attempts to authenticate itself to a system. The final case is where a potentially legitimate user presents ambiguous credentials. Deployment and testing of the scheme prove that soliciting user feedback is a very useful approach for accurate classification of spam messages.

The rest of the paper is organized as follows; Section II discusses work related to spam detection found in the literature. The two-tier spam detection scheme is presented in Section III. In Section IV, we provide the results obtained through experiments conducted on the Spam Assassin Corpus. We present our concluding remarks and future directions of work in Section V.

II. Related Work

Spam detection has remained a key domain of research for information security researchers for over three decades. Similar to intrusion detection systems, the two variables of most interest are the percentages of messages classified correctly and the rate of false positives. The former expresses how well a classifier works, whilst the latter is a measure of incorrectly classified ham messages. In this section, we highlight research findings on spam detection.

 ISBN: 978-1-61208-439-8

The use of machine learning for detecting spam has been studied and analyzed in [3]. A locally-acquired dataset was deployed for the experiments and a total of three popular classifiers, namely, k-Nearest Neighbors (k-NNs), Multi-Layered Perceptrons (MLPs) and Support Vector Machines (SVMs), were tested. The highest accuracy in spam detection was reported by the SVM classifier, with a 77% accuracy in message classification, at the cost of 22% false alarms.

Seminal work done on spam classification was through the use of a Naïve Bayesian classifier for detecting spam in [4]. Though basic characteristics of spam classifiers are common, the dictionary of spam words has grown significantly over time. In addition, the ability of spammers to circumvent existing controls has led to significant losses for businesses. In [5], a classification technique is presented for web spam, where web spam is defined as deliberate attempts to circumvent the results generated through a search query, when made by an end-user through a search engine of choice. The ranked list of query results are effectively populated with link-stuffed pages (having little or no relevant content) and keyword-stuffed pages (containing one or more keywords typed in by an end user). Classification of spam based on two sets of features, a baseline feature set and a query-independent/query-dependent feature set, was done using the SVM classifier. The results showed a 60% precision and a 10.8% recall for the baseline feature set. The recall rate improves significantly when the two feature spaces (page-level and rank-time) are combined.

The authors present an approach for identification of key attributes of an email header, in [6]. It is stated that header-message analysis is a superior option to message-body analysis, from a performance perspective. Email header fields such as message type, deliver status results and content descriptors are useful in differentiating spam from legitimate mail. The authors highlight the benefits of analyzing specific email header fields as opposed to others.

In [7], an ensemble-based learning technique is presented for detecting spam. The authors propose a framework for online spam detection through classification of labeled data into one of three classes, namely, self data, peer data and public data. Self data is collected from an individual user through explicit judgements and implicit judgements. A web browser plug-in provides an interface for the users to submit labels for spam. Judgements collected through browser-based user activity help produce a database of spam words that may be evolved over time. In addition, peer data for spam classification is also shared for construction of the spam database. The authors evaluate the proposed framework on the Web Spam dataset. Results obtained through the application of the Random Forest and Random Tree classifiers on the labels obtained through the ensemble framework, portrayed a 100% accuracy.

A feature selection method to detect spam accurately, is presented in [8]. The proposed scheme applies several association coefficients to the spam dataset, for generating similarity scores between the data found in a spam dataset and the messages being analyzed. The results obtained through application of these similarity measurement techniques portrayed a high success rate ($\sim$ 98%), for 6 out of 7 similarity computation methods.

In [9], several artificial intelligence techniques are tested on spam that targets short message service texts. Bayesian networks presented the highest accuracy in classification whereas attribute-based classification of messages portrayed the poorest performance.

Fusion of spam messages based on input from several fusion engines operating in parallel, is presented in [10]. The incoming stream of email is presented to a filter, which labels the message as being either spam or ham. The base filters operating in parallel produce a spamminess score and a binary classification for each message analyzed. The fusion of individual votes obtained from the binary base filters yields a fused score between 0 and 1, for decision-making purposes. Results obtained through experiments showed a 0.1% ham misclassification rate through score fusion.

Unlike the various approaches found in the literature for spam classification, the novelty in our proposed mechanism lies in its ability to re-classify messages that are originally classified as neither spam nor ham.

III. Proposed Scheme

Current spam filtering systems operate as follows: Spam words listed in a dictionary are compared against the words extracted from the incoming email message. The two-tier spam classifying scheme proposed in this paper introduces two key features to the typical spam classifier. The first is a mechanism for soliciting user feedback and the second is the Spinbox. The operation of the scheme is presented in Algorithm 1. The architecture of the scheme is illustrated in Figure 1.

The scheme is dependent on user feedback for training of the spam classifier on messages that neither fall into the spam nor the ham categories of messages. Traditional machine learning algorithms tend to classify messages as either ham or spam through static training during system initialization. Subsequently, retraining occurs only through re-initiation of the training process of the newer sets of spam words. As a result, the accuracy in spam classification is negatively affected. Moreover, the absence of an automated client-side mechanism for identifying and re-tagging words of the grey list into either spam or ham, remains a major hindrance to the performance of the spam classifier. Our proposal of a two-tier user feedback-based spam classifier provides a higher degree of accuracy in classification by assigning the decision-making task to the human user. The success of the proposed scheme lies in the variability in message classification across a range of human subjects. A message that is categorized as being spam by one user may be identified as being legitimate by another. Therefore, the presented scheme classifies messages based on feedback solicited from individual users.

Through inclusion of a user feedback-based mechanism, we incorporate human opinion in the decision-making process

 ISBN: 978-1-61208-439-8

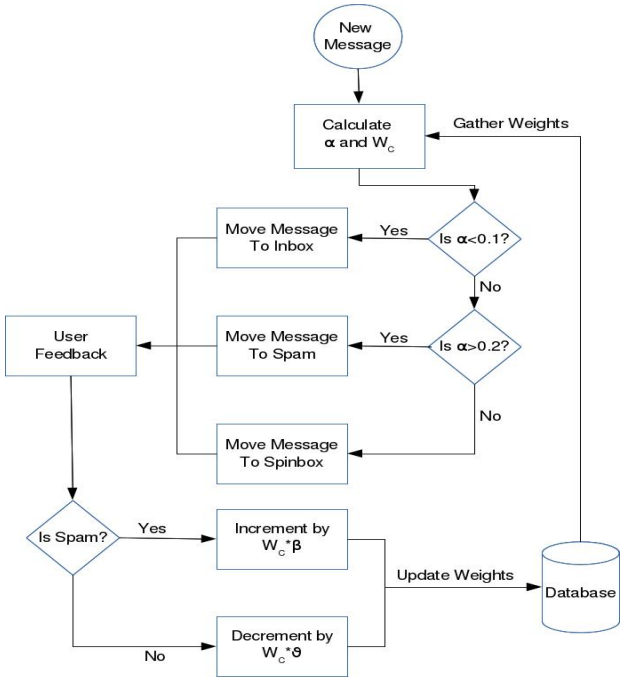

Fig. 1. The proposed two-tiered spam classification scheme.

allowing the system to make decisions on a per-user basis. Also, the system provides an opportunity for the user to mark certain words and/or phrases as spam rather than marking the whole message as spam. This implies that instead of learning "what" is spam, the system in addition also learns "why" a given message is indeed spam.

The purpose of the Spinbox is to provide a middle ground, i.e., grey area, between ham and spam message classes. This means that instead of marking an e-mail message as either legitimate or spam, it can be classified as undecided. In simpler terms, a message classified as undecided means that the contents of the message are classified differently by individual users. Therefore, the system cannot make a decision about the legitimacy of the message with the information at hand and thus needs further feedback. Not only does this improve the statistical accuracy of the system, but it can also be used to learn about the state of the system at any point in time. The lower the number of messages in the Spinbox, the better the system is trained to distinguish between spam and ham.

A. Two-tier Classification

The traditional spam classifier classifies a message through a binary classification of messages into either ham or spam. Considering the dual-class issue associated with binary classifiers, messages unclassified in clear terms pose a challenge from a statistical analysis viewpoint to the performance of the classifier. Our proposed solution addresses this problem and allows for a system to improve its accuracy by incorporating a second tier of classification, through categorizing of messages as grey and their subsequent placement into the Spinbox. Messages are thus classified as undecided when they meet certain criteria, rather than categorizing them into strictly ham or strictly spam.

1. *Message Preprocessing*
   Message is parsed and a word graph constructed.
2. *Weight Retrieval*
   For each word $j \in$ message N **do**:
   Retrieve word weight W[j] from Database
   Construct Y[] as a 1-D array of word weights
3. *Parameter Calculation*
for *i=1 to* $Length(Y)$ **do** **if** *Message[i]* $\in$ *Spam_List* **then** N_S++;
Calculate α ;
4. *Message Classification*
   if $\alpha < 0.1$ **then** Message = Legitimate;
   if $\alpha > 0.2$ **then** Message = Spam;
   if $0.1 \leq \alpha \leq 0.2$ **then** Message = Grey;

Algorithm 1: Two-Tier Spam Detector

B. Weight Assignment

The system works by assigning weights to words/phrases that are stored in the database. When a new message arrives, the system makes a decision to classify the message as ham, spam, or undecided based on the value of α which is calculated as shown in (1).

$$\alpha = \frac{N_s * W_T}{N_T} \tag{1}$$

where,
N_S = Number of spam words in a message
W_T = Cumulative weight of all words $\in Y$
N_T = Total number of words in the message

For the purposes of this experiment, the threshold for an undecided message was defined as the value of α between 0.1 and 0.2. Values lower than 0.1 were considered legitimate while values greater than 0.2 were classified as spam. If the cumulative score, α of the entire message, based on calculations through (1) and (2), leads to its classification into the grey class, the message is moved to the Spinbox. Once the system has classified the message, the next step comes in i.e., user feedback. The system prompts the user to provide categorization of the grey message into either a spam or a ham. Subsequently, the system adds the identified spam words to the database if they don't already exist or updates their weights otherwise. Updating of weights is done through the computation of W_C, which is calculated as follows.

$$W_C = \frac{\alpha}{N_S} \tag{2}$$

where W_C represents the calculated weight. When a word or phrase is classified as spam by the user, its weight is incremented by $W_C * \beta$. On the other hand, when a message is

 ISBN: 978-1-61208-439-8

classified as legitimate by the user, the weight for spam words present in that message is decremented by the value, $W_C * \theta$. The default values for β and θ used for the scheme were 1 and 2, respectively. This translates to "for every person claiming that a message is legitimate, there ought to be at least two people claiming it to be spam, in order to create reasonable doubt." A set of three different $\{\beta, \theta\}$ value pairs were tested as part of the experiments (see Section IV).

Three scenarios were studied as part of the proposed scheme:

Scenario 1: In this scenario, the values of β and θ were 1 and 2, respectively. This scenario was used with ideal values as a baseline to compare against subsequent scenarios. The results were expected to show a mix of upward and downward trends in weights assigned to the newly identified spam words, directly proportional to the incrementing feedback on spam words from end-users.

Scenario 2: This scenario represents a spam-tolerant system. It was used to test if the system would produce better results if there is higher tolerance to spam messages. Resulting performance was expected to pose fewer numbers of false negatives. For this scenario, the values of β and θ were chosen as 1 and 3, respectively. This translates to: for every instance of positive feedback for a grey message, the system requires three instances of negative feedback to classify a message as $undecided$. These values were selected to allow the system to tolerate misclassification of spam messages as opposed to the misclassification of hams. The results were expected to show a downward trend in the weight assignation, resulting in a spam-tolerant system.

Scenario 3: This scenario represents a high precision i.e., spam-intolerant system. It was used to test if the system would perform better in terms of spam detection if it posed a higher spam intolerance. For this purpose, the values assigned to β and θ were 3 and 1, respectively. This translates to: for every single instance of negative feedback, the system requires three instances of positive feedback to classify a message as $undecided$. These values were chosen to make the system more rigid without focusing too much on ham. The system was expected to show low tolerance to spam thus projecting an upward trend in the weight assignation.

IV. Results and Analysis

The experiments were conducted on a Linux machine with 16GB RAM and an AMD FX-8150 octa-core CPU. The dataset adopted for testing the performance of the proposed scheme was the Spam Assassin Public Corpus [11]. This dataset comprises of 1897 spam messages, all obtained through non-spam-trap sources. It also includes 3900 easy-ham non-spam messages. These messages are easily differentiable from spam since they rarely contain spam signature words. The dataset also contains 250 non-spam hard ham messages, defined as being similar to spam, but falling in a different class altogether. Hard ham messages use HTML tags, irregular HTML markup tags, coloured text, and phrases that appear to be spam.

In our work, we use three metrics for evaluating the proposed spam detector, namely, precision, recall and accuracy. Precision is defined as the fraction of correctly classified spam messages from the total number of messages analyzed, given by $\frac{TN}{TN+FN}$, where TN represents true positives and FN represents false negatives. Recall, on the other hand, is the total number of correctly classified spam messages over the total number of spam messages found by the system, $\frac{TN}{TN+FP}$. The accuracy is given by, $\frac{TP+TN}{TP+TN+FP+FN}$. The experiment was designed to run in an iterative manner. The system performed its classifications through k iterations, with each iteration representing a single user feedback. For instance, if the value of k is equal to 10, it means that the system has received feedback on a single grey message from a total of 10 users. All experiments were run with k = 10.

Table I shows the values for precision, recall, and accuracy of the system based solely on user feedback without the Spinbox in place, whereas table II shows the results with the Spinbox included. Even though scenarios 1 and 2, portray similar values for precision, recall and accuracy, both with and without a Spinbox in place, we can notice a clear difference in the results obtained for scenario 3. This is because scenario 3 was designed to be more spam-intolerant than the other two scenarios, and therefore, the grey messages were categorized as spam during the initial iterations of the spam classification process. It is safe to assume that a variation in scenarios 1 and 2 would have been evident provided that more user feedback was considered in the grey message detection step. The false negatives portrayed in the tables are final values after passing through a number of user feedback iterations (equal to the value of k). As a result, the initially-generated false negatives converged to zero after completion of the k stipulated iterations. It can also be concluded based on the data shown in these tables that adding the Spinbox improved the accuracy and recall of the system and presented a marginal improvement in the precision.

Figure 2 shows the system's true and false negative trends after 1, 5, and 10 iterations, respectively, for all three scenarios. The results clearly depict that scenario 1 yielded expected results. Scenario 2, however, showed similar results to scenario 1, if observed, instead of showing a spam-tolerant behaviour. A detailed investigation of the weights stored in the database revealed that the results of both the scenarios had a lot of variance and scenario 2 was in fact following a downward trend in weight assignation. Given enough inputs and feedback, it is safe to say that at a certain point in time, the system would have allowed more numbers of spam messages to be classified as legitimate. Scenario 3 performed as per expectation, and showed an extreme intolerance to spam even in the early stages (smaller k values). However, this scenario had a huge drawback. It went overboard with its intolerance because of its rigid characteristics. After several iterations (incrementing k values), the system ended up marking legitimate messages

 ISBN: 978-1-61208-439-8

as spam even for the most marginal hints of spam. Figure 3 shows the true and false positive trends of the system. As it is evident, scenarios 1 and 2 were able to classify legitimate messages accurately, with minimal false positives. Scenario 3, on the other hand, had an upward trend of false positives due to its low tolerance to spam messages. The second major component of this whole system was the Spinbox, which contained messages that were classified as $undecided$. Figure 4 shows the system's trends for undecided messages for all three scenarios. It was clear that as user feedback increased, the system was better able to classify a message resulting in a lower number of undecided messages. After k iterations, the system was only unable to classify 2.5% of the messages. This number does change with increasing values of k.

We also ran the same tests using only the user feedback without the Spinbox factor. Figures 5 and 6 show the True and False negatives and True and False positives without the Spinbox in place. The results clearly showed that adding the Spinbox reduced the number of false positives and negatives thus increasing the accuracy of the system.

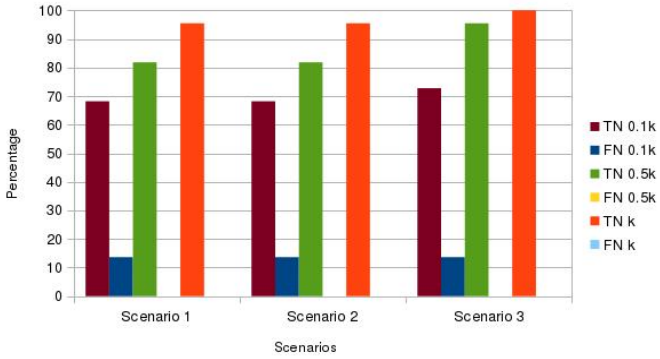

Fig. 2. True Negatives and False Negatives for the Three Scenarios with Spinbox.

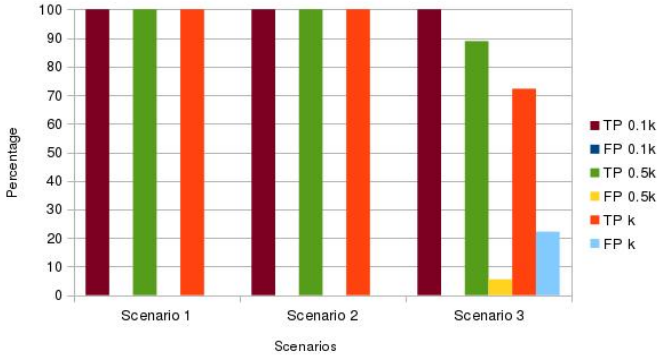

Fig. 3. True Positives and False Positives for the Three Scenarios with Spinbox.

In Table III, we compare the results obtained from the three scenarios tested, with other popular schemes found in the literature, for the same dataset. As is evident from the results, the performance of the proposed scheme outclasses the four

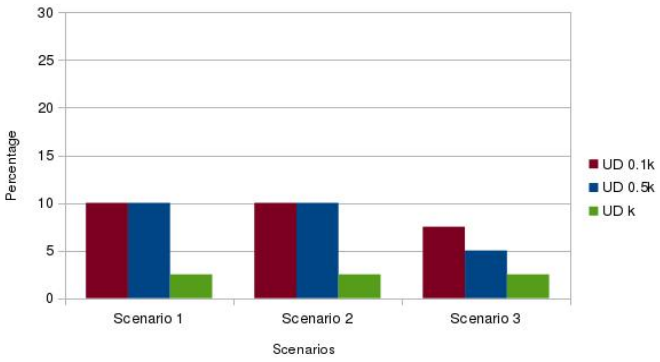

Fig. 4. Undecided Trend for the Three Scenarios with Spinbox.

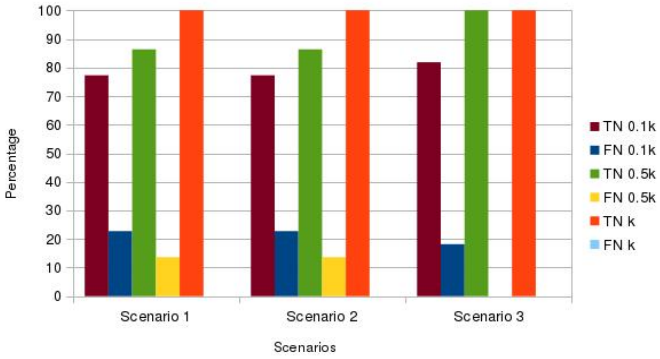

Fig. 5. True Negatives and False Negatives for the Three Scenarios without Spinbox.

other techniques, namely, multi-layered perceptrons (MLPs), ranked time features, ensemble-based classifiers, and correlation coefficient based feature ranking and selection. Some of these techniques do not have values reported for specific performance measurement metrics. For instance, MLP does not have a reported precision value. Overall, the feedback-based mechanism proposed does effective classification of spam messages, and is therefore viable for deployment in a production environment.

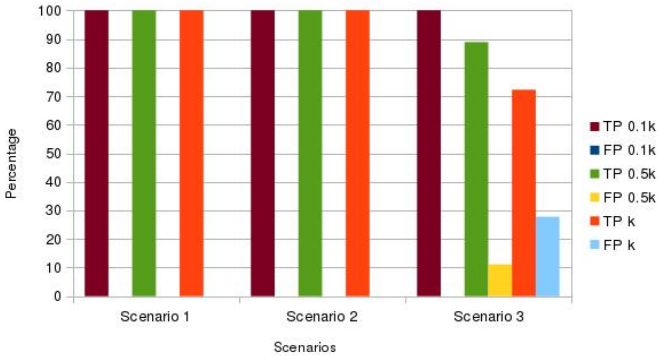

Fig. 6. True Positives and False Positives for the Three Scenarios without Spinbox.

 ISBN: 978-1-61208-439-8

TABLE I. PERFORMANCE METRICS FOR SCHEME WITHOUT SPINBOX

	True Positive	False Positive	True Negative	False Negative	Precision	Recall	Accuracy
Scheme 1	18	0	22	0	1	1	1
Scheme 2	18	0	22	0	1	1	1
Scheme 3	13	5	22	0	1	0.815	0.875

TABLE II. PERFORMANCE METRICS FOR SCHEME WITH SPINBOX

	True Positive	False Positive	True Negative	False Negative	Precision	Recall	Accuracy
Scheme 1	18	0	21	0	1	1	1
Scheme 2	18	0	21	0	1	1	1
Scheme 3	13	4	22	0	0.765	0.846	0.897

TABLE III. A COMPARISON OF RESULTS WITH OTHER SCHEMES

	MLP [3]	Rank Time Features [5]	Ensembles [7]	Feature Selector [8]	Scenario 1	Scenario 2	Scenario 3
Precision	–	0.6	–	–	1	1	1
Accuracy	0.93	–	1	0.98	1	1	0.897
Recall	–	0.11	–	–	1	1	0.846

V. CONCLUSION AND FUTURE WORK

Classifying spam through accurate analysis by automated classifiers has produced less-than-acceptable performance levels. We have presented a two-tiered user feedback-based approach for accurately classifying emails as spam. The results obtained through experiments showed promise. For any system to be good at spam detection, it has to be able to adapt to changing paradigms i.e., must evolve alongside corresponding evolution of the spammer class. By incorporating user feedback into the spam classification process, we not only empower the user but also ensure that the system does online tagging of messages that can be categorized as neither ham nor spam. The proposed scheme does spam classification through solicitation of user feedback on messages tagged as being grey, through analysis of all words found in the message. We acknowledge, however, that asking users to classify large volumes of words may be impractical in some applications.

As part of our future work, we intend to test the proposed spam classifier on diverse publicly-available datasets. In addition, we shall be proposing a machine learning-based scheme to automatically generate scores on incoming grey messages, and fuse the same with scores obtained from user feedback. The resulting scheme is expected to improve the accuracy of the spam classifier.

We plan to extend this work to authorization systems by incorporating this scheme into an XACML-based ontology mapper. This has obvious security benefits as fraudulent (adversary) users will not be able to present partially correct URIs to spoof access to other systems.

ACKNOWLEDGMENT

This work has been partially funded by the European Commission via grant agreement no. 611659 for the AU2EU FP7 project.

REFERENCES

[1] T. Technology, "What is the real impact of spam on business?" http://www.topsectechnology.com/it-security-news-and-info/what-is-the-real-impact-of-spam-on-business, Dec 2014.

[2] TheEmailAdmin, "Do you trust your bank not to spam you? read this," http://www.theemailadmin.com/2014/03/do-you-trust-your-bank-not-to-spam-you-read-this/, Mar 2014.

[3] R. Lakshmi and N. Radha, "Spam classification using supervised learning techniques," in *A2CWiC, 2010 Conference on*, Sep 2010.

[4] P. Pantel and D. Lin, "Spamcop: A spam classification & organization program," AAAI, Tech. Rep., 1998.

[5] K. Svore, Q. Wu, C. Burges, and A. Raman, "Improving web spam classification using rank-time features," in *AIRWeb, 2007 Conference on*, May 2007, pp. 9–16.

[6] S. bin Abd Razak and A. Bin Mohamad, "Identification of spam email based on information from email header," in *Intelligent Systems Design and Applications (ISDA), 2013 13th International Conference on*, Dec 2013, pp. 347–353.

[7] C. Dong and B. Zhou, "An ensemble learning framework for online web spam detection," in *Machine Learning and Applications (ICMLA), 2013 12th International Conference on*, vol. 1, Dec 2013, pp. 40–45.

[8] A. Abdelrahim, A. Elhadi, H. Ibrahim, and N. Elmisbah, "Feature selection and similarity coefficient based method for email spam filtering," in *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, Aug 2013, pp. 630–633.

[9] K. Mathew and B. Issac, "Intelligent spam classification for mobile text message," in *Computer Science and Network Technology (ICCSNT), 2011 International Conference on*, vol. 1, Dec 2011, pp. 101–105.

[10] T. Lynam and G. Cormack, "On-line spam filter fusion," in *SIGIR, 2006 Conference on*, Aug 2006, pp. 123–130.

[11] S. Assassin, "Spam assassin public corpus," https://spamassassin.apache.org/publiccorpus/, Tech. Rep.

 ISBN: 978-1-61208-439-8