Comparison of anomaly detection-based spam email filters

Aleksej Miloševič

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Smetanova 17, 2000 Maribor

aleksej.milosevic@student.um.si

Napredek komunikacijske tehnologije vsakodnevno povezuje veliko število ljudi, vendar prav tako botruje čedalje večji količini odvečne elektronske pošte. Napadalci razvijajo napredne metode izogibanja filtrom odvečne pošte, na katere je potrebno odgovoriti z izpolnjevanjem in dopolnjevanjem obstoječih tehnik filtriranja. V članku predstavimo zaznavanje odvečne pošte z algoritmi zaznavanja anomalij osnovanim na strojnem učenju, katere glavna prednost je zmanjšana potreba po pred-označenih podatkih. V okviru raziskave smo primerjali tri algoritme zaznavanja anomalij: Simple Anomaly Detection Method, One-Class SVM in Isolation Forest. Ugotovili smo, da je optimalna metoda zaznavanja odvečne elektronske pošte odvisna od podatkovne množice in izbire bodisi legitimnih bodisi odvečnih dokumentov za zastopanje normalnosti.

With the widespread availability of broadband internet and affordable connected devices, spam email is a more prominent threat than ever before, incurring billions of dollars in costs yearly. Attackers use increasingly advanced methods of spam filter evasion techniques which forces the cybersecurity community to respond with innovative techniques to supplement existing countermeasures. One such technique is machine learning-based anomaly detection which reduces the need for pre-labelled data. In this paper we compare the effectiveness of three anomaly detection methods: Simple Anomaly Detection, One-Class SVM and Isolation Forest. We have found that the optimal method of anomaly detection depends on the dataset and the choosing legitimate or spam documents as representation of normality.

1 Introduction

Electronic mail or e-mail is one of the most important and useful channels of communication that exist today because it enables cheap and efficient communication between interested parties with zero overhead to the participants. Spam, according to Geerthik is “an endless repetition of worthless text or image”(Geerthik, 2013). Email can be easily exploited because of the inherently unsafe nature of the Simple Mail Transfer Protocol or SMTP. The protocol has no internal mechanism to secure the contents and routing or to validate the sender (Geerthik, 2013)

According to the Federal Trade Commission, two thirds of all emails contain misleading information. The dangers of spam ranges from a minor annoyance such as having to delete the email landing unrestricted to an individual’s inbox, to massive economic risks. According to calculations made by Rao and Reiley spam email costs between $14-18 billion per year while only generating a revenue of $160-360 million per year (Rao & Reiley, 2012).

Mitigating spam mail is an evolving problem that requires innovative solutions. To this date social methods of fighting spam have proven cumbersome and ineffective and as such technical countermeasures were developed to ensure efficient protection. Before the rise of machine learning based spam filtering. filters followed a so called *‘knowledge engineering’* approach which utilized a content-based heuristic filter. Such filters analyzed messages and classified them as either ‘*spam’* or *‘ham’* based on the knowledge of regularities or certain observed patterns such as those defined by Guzella and Caminhas. (Guzella & Caminhas, 2009).

Heuristic filters, however, have a considerable drawback in the form of cumbersome and labor intensive updating of the rule-set in order to keep up with the newest trends. An example of such trends is the obfuscation of popular terms with special characters (e.g. “V!@g®@” instead of “Viagra”). Filters are supplemented by a plethora of other mitigation methods. Blacklisting involves keeping a list of e-mail or IP addresses of servers from which spam mail is likely to originate. A whitelist is the natural reverse of blacklists where an email server or a client keeps a list of trusted e-mail addresses or server IP’s that are allowed communicate with the user. If an incoming message is from an IP or email address that is neither on the blacklist or the whitelist, the recipients email server may temporarily reject the message. If the email is in fact legitimate, the sender will attempt, after some time. to send the message again upon which the recipient will accept the email as legitimate and add the sender’s credentials to a whitelist (Bhowmick & Hazarika, 2016).

Modern spam filters have evolved to use machine learning predictive models which are by their very nature adaptive and require less labor and manpower to maintain. Most machine learning and statistical approaches to filtering spam are supervised. This requires a prelabelled training set which is used to train an algorithm and build a predictive model. It is imperative that the number of instances in the training set is enough to reach satisfactory model accuracy. The biggest problem utilizing supervised predictive models is obtaining the large amount of labelled data necessary (Santos, Laorden, Ugarte-Pedrero, Sanz, & Bringas, 2011).

In this paper we compare three anomaly detection-based solutions. Section 2 describes in detail the structure of text-based spam filters. Section 3 describes the implementation of a vector space model anomaly detection method. Section 4 describes the execution of the experiment and displays the generated data and compares and discusses classification results of three methods; implemented vector space model anomaly detection method, isolation forest and one-class SVM. The fifth and final section concludes the paper and proposes further research.

2 Spam filter structure

An email message is divided into a header, which contains the metadata of an email and the body which contains the actual message of the sender to the receiver. Before executing classification, the contents of the email need to be pre-processed in a form that is appropriate as input to the classifier. The pre-processing is conducted in the following steps (Bhowmick & Hazarika, 2016) :

1. Tokenization, which is the process of extracting individual words from the email
2. Lemmatization, which reduces the words to their root form
3. Stop-word removal, which removes the most common words in the English language (e.g is, and, the)
4. Representation conversion, which converts words into a machine-readable format

2.1 Vector space model representation

Emails are represented in the form of an Information Retrieval model. Specifically. we use a term frequency-inverse document frequency, or tf-idf model.

The tf-idf is a product of term frequency and inverse document frequency. This is calculated by using the number of times a term *t* occurs in document *d*, where the term frequency of term *j* in document *i* is defined as (Laorden et al., 2014) :

$$tf_{i,j} = \frac{m_{i.j}}{\sum_k m_{k,j}}$$

- $m_{i,j}$ is the number of times the word $t_{i,j}$ appears in a document
- $\sum_k m_{k,j}$ is the number of words in a document

The inverse document frequency is defined as:

$$idf_i = \frac{|\varepsilon|}{|\varepsilon : t_i \in e|}$$

- $|\varepsilon|$ is the total number of documents containing the word $t_{i,j}$
- $|\varepsilon : t_i \in e|$ is the number of documents containing the word $t_{i,j}$

The term frequency – Inverse document frequency is then calculated as:

$$tfidf\ (i,j) = tf_{i,j} * idf_i$$

3 Anomaly detection

Anomaly detection is the process which enables finding of patterns in data that does not conform to expected behavior. As such, anomalies are patterns that do not conform to a notion of what is considered normal behavior (Chandola, Banerjee, & Kumar, 2007). The precise method of detecting anomalies differs from method to method.

An isolation forest explicitly identifies anomalous data points. Each tree conducts its search by isolating observations based on a randomly selected feature and randomly selected split value between the maximum and minimum of the feature. If individual trees collectively produce shorter path lengths, they are likely to be anomalies. The score anomaly score is defined as (Liu & Ting, 2018):

$$s(x,n) = 2^{-\frac{E(h(x))}{c(n)}}$$

- *h(x)* is the path length of observation *x*
- *c(n)* is the average path length of unsuccessful search in a Binary Search Tree
- *n* is the number of external nodes

If the individual score of an observation is close to 1 it strongly indicates an anomaly. If the individual score is low, it indicates a normal observation. Uniform scores around 0,5 indicate that the observations are relatively uniformly distributed and that there are no clear anomalies.

A One-Class SVM is a variant of the Support Vector Machine algorithm which defines a hyperplane that separates normal data from anomalous data. The binary function captures a region in the area of a certain high enough probability density, using the chosen kernel function. For each point, the function returns +1 in a small region, which represents the training data points and -1 elsewhere. (Manevitz, 2001)

3.1 Simple anomaly detection method

Our implemented method as described in detail by Laorden et. al in the paper *Study on the effectiveness of anomaly detection for spam filtering*, analyses points in a feature space by extracting features from the document and measuring the distance from the point representing the email to the points that symbolize normality. Normality is represented by the model created using the training data. The distance between two points is defined by Euclidean distance (Laorden et al., 2014) :

$$d(x,y) = \sum_{i=0}^{n} \sqrt{x_i^2 - y_i^2}$$

- *x* is the first point
- *y* is the second point
- x_i is the *i*-th component of *x*
- y_i is the *i*-th component of *y*

The result of the training method is a final distance value which considers every measure performed. The paper recommends three distinct metrics, also called combination rules:

- The mean value calculated from every distance in the training set
- The lowest distance value from every distance value in the training set
- The highest distance from every distance value in the training set

The algorithm establishes 10 different thresholds with which to determine whether the email is spam or not. The thresholds are chosen by first establishing the lowest and the highest one. The lowest threshold is determined as the lowest possible value at which no legitimate spam messages were misclassified. The highest threshold is determined as the lowest possible value at which no examples of legitimate mail were misclassified. For each of the thresholds three metrics are calculated; False Negative Ratio (FNR), False Positive Ratio (FPR) and the Weighted Accuracy (WA). False Negative Ratio is defined as (Demsar, 2006):

$$FNR(\beta) = \frac{FN}{FN + TP}$$

- FN is the number of documents misclassified as legitimate
- TP is the number of documents correctly classified as spam

False Positive Ratio is defined as:

$$FPR(\beta) = \frac{FP}{FP + TN}$$

- FP is the number of documents misclassified as spam
- TN is the number of correctly classified legitimate messages

Weighted Accuracy is defined as:

$$WA(\beta) = 1 - \frac{FNR + FPR}{2}$$

4 Experiment and results

We have utilized data from the LingSpam Corpus and EnronSpam public Corpus. The LingSpam Corpus contains messages originated from the *Linguistic list,* which is an email distribution about linguistics. The EnronSpam public corpus contains messages originating from the senior management of the now defunct company Enron. We have opted for the ‘bare’ datasets as opposed to the pre-processed ones. Due to computational limitations we have trimmed the dataset to a manageable size as seen in the following table:

Table 1 Number of documents in dataset per type

	LingSpam	EnronSpam
# of legitimate emails	2412	8033
# of spam emails	481	2966

We have removed alpha-numeric and non-letter characters on both datasets. We have then performed stop word removal and stemming and transformed the datasets into a tf-idf matrix. For each dataset we have performed 10-fold Monte Carlo cross-validation in the ratio of 0.66:0.33 in favor of the training set (Tech & Gomez-gil, 2013). We have performed two sets of tests. The first set of models was trained on ‘ham’ emails with spam emails being regarded as anomalies. The second set of models was trained on spam emails with legitimate emails being regarded as anomalies.

4.1 Results

In the following subchapter the results of the described experiments are presented. The results are presented on a per-dataset basis, sorted by the training class (ham or spam) and the trained algorithm. For Isolation Forest and One-Class SVM we have used the average value aggregated from the 10 folds. For the Simple Anomaly Detection method, we have used the data from the threshold with the highest weighted accuracy.

Table 2 shows the results of the models trained on the LingSpam algorithm. Trained on legitimate documents, the worst performing algorithms are One-Class SVM and Simple Anomaly Method – MAX which have weighted accuracies near chance. The best algorithm is the Simple Anomaly Method – MIN, which has a weighted accuracy of 0,78. This method has a false negative ratio of 0,04 and a false positive ratio of 0,4. This means that the model is seldom wrong when it comes to identifying legitimate mail, but has some issues determining whether an email is spam. In the case of LingSpam, the results are markedly better when models are trained on spam email. As before, Simple Anomaly Detection – MAX fails at a weighted accuracy of near chance – 0,56. The best performing algorithm was the One-Class SVM with a weighted accuracy of 0,9. The false negative ratio is 0 and the false positive ratio is 0,55. This means that the method never misclassifies a spam document as a ham document.

Table 2 Lingspam Corpus results

LingSpam	Algorithms				
Weighted Accuracy	Isolation Forest	One-Class SVM	Simple Method-MAX	Simple Method - MEAN	Simple Method - MIN
Trained on 'ham'	0,685	0,522	0,5	0,738	0,78
Trained on 'spam'	0,805	0,9	0,56	0,861	0,86
False Negative Ratio					
Trained on 'ham'	0,798	0,165	0	0,242	0,04
Trained on 'spam'	0,04	0	0,009	0,083	0,28
False Positive Ratio					
Trained on 'ham'	0,138	0,521	1	0,28	0,4
Trained on 'spam'	0,925	0,55	0,88	0,19	0

Table 3 shows the results of the models trained on the EnronSpam algorithm. As with LingSpam the worst performing methods trained on ‘ham’ are One-Class SVM and Simple Method – MAX. The best performing method is Simple Method – MIN with a weighted accuracy of 0,92. Both the false negative and false positive ratios are excellent at 0,05 and 0,01 respectively. Trained on ‘ham’ documents, the worst performing algorithm is Simple Anomaly Detection method – MAX with a weighted accuracy of 0,5. The best performing method is again Simple Method Min with a FNR of 0,5 and a FPR of 0,3. In general, the weighted accuracies are high despite the sometimes mediocre FNR and FPR because of the imbalanced dataset.

Table 3 EnronSpam Corpus results

EnronSpam	Algorithms				
Weighted Accuracy	Isolation Forest	One-Class SVM	Simple Method-MAX	Simple Method - MEAN	Simple Method - MIN
Trained on 'ham'	0,683	0,621	0,65	0,83	0,92
Trained on 'spam'	0,685	0,791	0,5	0,69	0,83
False Negative Ratio					
Trained on 'ham'	0,801	0,028	0,06	0,09	0,05
Trained on 'spam'	0,088	0,089	0	0,17	0,05
False Positive Ratio					
Trained on 'ham'	0,131	0,501	0,63	0,23	0,01
Trained on 'spam'	0,898	0,535	1	0,44	0,3

To determine the superior method of training the models we have compared the average and maximum weighted accuracies for both corpora as shown in Table 4. On average, the LingSpam corpus did better trained on ‘spam’ with an average weighted accuracy of 0,792 versus the accuracy of 0,645 when trained on ‘ham’ documents. This is also reflected in the maximum weighted accuracies of 0,9 for ‘spam’ and 0,78 for ‘ham’. The opposite is true for the EnronSpam corpus. Trained on ‘ham’ documents, the average weighted accuracy of the models is 0,74 compared to 0,69 when trained on ‘spam’ documents. This trend continues with the maximum weighted accuracies of 0,92 for the best performing model trained on ‘ham’ documents and 0,83 for the best model trained on ‘ham’. We can recognize from this that the training method depends on the dataset.

Table 4 Average and maximum weighted accuracies by type of training

Average Weighted Accuracies	LingSpam	EnronSpam
Trained on 'ham'	0,645	0,7408
Trained on 'spam'	0,7972	0,6992
Maximum Weighted Accuracies	LingSpam	EnronSpam
Trained on 'ham'	0,78	0,92
Trained on 'spam'	0,9	0,83

5 Conclusions

Continuously improving spam detection techniques is crucial to the effort of continuously improving safety on the internet. Increasingly sophisticated filter evasion methods demand an increasingly larger stack of technologies, each approaching the problem of spam detection from a different perspective.

In this paper, we have approached this problem from an anomaly detection perspective. We have implemented the Simple Anomaly Detection Method proposed by Laorden et. al and compared the resulting weighted average, false negative ratio and false positive ratio with the Isolation Forest and One-Class SVM algorithms. We have built separate models with legitimate and spam documents as representations of normality. We have found the results to be satisfactory, with the best method being the Simple Anomaly Detection using the minimum combination rule.

There are many avenues of research that are yet to be explored in this area. First and foremost, the research in this paper should be furthered by utilizing additional metrics like the ROC curve and AUC (Demsar, 2006). It would also be prudent to incorporate larger datasets and additional corpora to broaden the understanding of the cause and effect of dataset size on classification performance. Secondly, we should consider further research on anomaly detection ensembles where each member of the ensemble specializes in detecting types of anomalies. Thirdly, we should consider evaluating the performance of anomaly detection systems after utilizing with white-listing, grey-listing and black-listing and before using supervised methods.

6 References

Bhowmick, A., & Hazarika, S. M. (2016). E-Mail Spam Filtering: A Review of Techniques and Trends. In *Advances in Electronics , Communication and Computing* (pp. 583–590). https://doi.org/10.1007/978-981-10-4765-7

Chandola, V., Banerjee, A., & Kumar, V. (2007). *Anomaly Detection: A Survey*. *Information Sciences journal* (Vol. 32). https://doi.org/10.1038/nn.2116

Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research 7*. https://doi.org/10.1016/j.jecp.2010.03.005

Geerthik, S. (2013). Survey on Internet Spam : Classification and Analysis. *International Journal of Computer Technology & Applications*, *4*(June), 384–391.

Guzella, T. S., & Caminhas, W. M. (2009). A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2009.02.037

Laorden, C., Ugarte-Pedrero, X., Santos, I., Sanz, B., Nieves, J., & Bringas, P. G. (2014). Study on the effectiveness of anomaly detection for spam filtering. *Information Sciences*. https://doi.org/10.1016/j.ins.2014.02.114

Liu, F. T., & Ting, K. M. (2018). Isolation Forest. In *Eighth IEE International Confrence on Data mining*. https://doi.org/10.1109/ICDM.2008.17

Manevitz, L. M. (2001). One-Class SVMs for Document Classification. *Journal of Machine Learning*, *2*, 139–154.

Rao, J. M., & Reiley, D. H. (2012). The Economics of Spam. *Journal of Economic Perspectives*, *26*(3), 87–110. https://doi.org/10.1257/jep.26.3.87

Santos, I., Laorden, C., Ugarte-Pedrero, X., Sanz, B., & Bringas, P. G. (2011). Anomaly-based Spam Filtering. In *Proceedings of the 6th International Conference on Security and Cryptography SECRYPT* (p. In press). Retrieved from http://paginaspersonales.deusto.es/bosanz/publications/pdf/2011/Santos_2011_SECRYPT_Anomaly-based_Spam_Filtering.pdf

Tech, Y., & Gomez-gil, P. (2013). An assessment of ten-fold and Monte Carlo cross validations for time series forecasting, (April 2015). https://doi.org/10.1109/ICEEE.2013.6676075