Improved Spam Detection using DBSCAN and Advanced Digest Algorithm

Alaa H. Ahmed
Faculty of Engineering
Islamic University Of Gaza
Gaza Strip, Palestine

Mohammed Mikki
Faculty of Engineering
Islamic University Of Gaza
Gaza Strip, Palestine

ABSTRACT

E-mail is one of the most popular and frequently used ways of communication due to its worldwide accessibility, relatively fast message transfer, and low sending cost. Nowadays, detecting and filtering are still the most feasible ways of fighting spam emails. There are many reasonably successful spam email filters in operation. The identification of spam plays an important role in current anti-spam mechanism.

For improving the accuracy of spam detection, an improved Filtering technique is presented which is based on the Improved Digest algorithm and DBSCAN clustering algorithm.

Using this technique, mails are represented using improved digest algorithm and then clustered using DBSCAN clustering algorithm. All similar emails which always categorized as spam are identified and clustered together where good mails that don't look similar like other mails are not clustered. This method greatly improves the filtering accuracy against latest proposed algorithms by 30 % and improves the resistance of spam detection against increased obfuscation effort by spammers, while keeping miss-detection of good emails at a similar level of older filtering methods.

General Terms

Spam Filtering, Security, Nilsimsa, Data Clustering, Collaborative Spam Filtering

Keywords

DBSCAN, Digest Based, Nilsimsa, spam, clustering.

1. INTRODUCTION

The use of internet has been extensively increasing over the past decade and it continues to be on the ascent. Hence the Internet is gradually becoming an integral part of everyday life. Internet usage is expected to continue growing and e-mail has become a powerful tool intended for idea and information exchange. Negligible time delay during transmission, security of the data being transferred, low costs are few of the multifarious advantages that e-mail enjoys over other physical methods. However there are few issues that spoil the efficient usage of emails. Spam email is one among them [1].

The term spam is used to describe any "unwanted" thing. Email spam is a set of unwanted electronic spam mail that contains nearly identical messages sent to huge number of recipients. Spam mail can be not only annoying but also dangerous to recipients. Clicking on links contained in spam emails may send users to phishing and malware .It also may include malware as scripts or other risky executable file attachments.

The problem of spam or Unsolicited Bulk Email (UBE) is becoming a pressing issue [2]. Spam email characterized by three main features:

•**Anonymity**: The address and identity of the sender are concealed

•**Mass Mailing:** The email is sent to large groups of people

•**Unsolicited:** The email is not requested by the recipients.

While no effective and complete solution to the spam problem is currently available, several moderately successful anti-spam techniques have been proposed, each operating along a different line. Here we present a shortlist of desired filtering techniques.

List-Based Filters: List-based filters attempt to stop spam by categorizing senders as spammers or trusted users, and blocking or allowing their messages accordingly. Senders in blacklist are considered spammers and all mails sent by them are blocked, where senders in whitelist are trustees and all mails sent by them are allowed.

Content-Based Filters: Rather than enforcing across-the-board policies for all messages from a particular email or IP address, content-based filters evaluate words or phrases found in each individual message to determine whether an email is spam or legitimate.

A word-based spam filter is the simplest type of content-based filter. Generally speaking, word-based filters simply block any email that contains certain terms.

Heuristic (or rule-based) filters like Spam Assassin [3] take things a step beyond simple word-based filters. Rather than blocking messages that contain a suspicious word, heuristic filters take multiple terms found in an email into consideration.

Bayesian filters employ the laws of mathematical probability to determine which messages are legitimate and which are spam. In order for a Bayesian filter to effectively block spam, the end user must initially "train" it by manually flagging each message as either junk or legitimate. Over time, the filter takes words and phrases found in legitimate emails and adds them to a list; it does the same with terms found in spam.

Collaborative Content Filtering: An important feature of spam, which can be exploited for detecting it easier, is its bulkiness. A spam bulk mailing consists of many copies of the same original spam message, each sent to a different recipient or group of recipients. The different copies from the same bulk are usually obfuscated, i.e. modified a bit in order to look different from each other.Spammers apply obfuscation in order to make collaborative spam detection more difficult. Indeed, in collaborative spam detection it is important to have a good technique for determining which emails belong to the same bulk. 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. If the collaborative spam detection is based purely on the evaluation of bulkiness, each recipient must be equipped with white lists of all the bulky sources from which she or he wants to receive emails.

2. BACKGROUND

The clustering of emails is done by two steps: getting the digests of emails and clustering the digests. In this section, we will introduce some digest algorithms used in anti-spam field and the DBSCAN clustering algorithm of similar data.

Digest Algorithm

There are many digest algorithms in the anti-spam field. In the distributed anti-spam mechanism DCC [4] (Distributed Checksum Clearinghouse), there are two digests Dig l and Dig 2 for each email. Dig l is the MD5 value of the email body after removing the simple characters such as comma and semicolon, etc. Dig 2 is the MD5 value of the words set which is composed of special words in the email. Using the MD5 algorithm can ensure different emails to have different digests, but it can't do well with the usual spam attack strategy. For Dig l, if the spam attacker adds some additional information in the email, the Dig l will be entirely different. For Dig2, if the spam attacker exchanges the positions of some sentences in the email, the Dig 2 will be entirely different. So the digest algorithms in the DCC mechanism aren't strong enough to be used in anti-spam field. The CTPH [5] is a text digest algorithm which is based on fragments hash. This algorithm divides the text into fragments first, and then calculates the hash values of all the fragments, finally gets a character string composed of the hash value as the digest. CTPH determines the similarity of the two texts by computing the edit distance of the digests. The CTPH algorithm can identify the similar texts accurately with editing differences, so it has been widely used in computer forensic and anti-spam field. However, this algorithm doesn't do well with the usual spam attack strategy neither. Adding special characters after some sentences can make CTPH digests of similar emails completely different.

The Nilsimsa algorithm used in DHTnil [6] is a local sensitive hash function. This algorithm generates 256 integer values by analyzing the text, and then gets the average value of these 256 integer values. For each integer value, if it is larger than or equal to the average value, the corresponding bit is 1. Otherwise, the corresponding bit is 0. At last, for each text we generate a 256 bits (32 bytes) digest. In order to determine the similarity of two texts, we need to compute the distance of the two Nilsimsa digests. The distance is defined as the numbers of bits with the same value in the same position of the two digests.

Clustering Algorithm

To cluster the email digests, first we need to know the three main features of the digests in the space: (1) digest of the spam is gathering over digest space, (2) shape of the digest subspace is unknown, and (3) digests of regular emails are distributed over digest space.

In view of the fact that clustering methods of data mining have a good performance in clustering, in this paper we adopt the DBSCAN algorithm[7] to cluster mail digest subspace. The clustering algorithms of data mining mainly include five types: partitioning methods, hierarchical methods, grid-based methods, model-based methods and destiny-based methods. According to the three main features of the digests mentioned above, we can see that the digest density plays an important role in distinguishing between regular emails and the spam, which is also important in the spam classification. So the density-based clustering algorithm DBSCAN is a good choice to cluster the emails. DBSCAN (Density-Based Spatial Clustering of Application with Noise) algorithm is a density-based classification algorithm, which makes the high density area clustered and can find the clusters with arbitrary shape in the space with noise nodes removed. The DBSCAN algorithm has three main features: (1) basing on the density, (2) can identify clusters of arbitrary shape, and (3) can identify noise nodes. These features entirely meet the three features of email digests. Therefore, DBSCAN is a more applicable clustering algorithm in the field of anti-spam.

3. RELATED WORK

Damiani [8] et al. ("open-digest paper") is well known and often cited for its positive findings about the properties of a digest-based collaborative spam detection technique. The technique produces similar digests out of similar emails, and uses them to find out which emails belong to the same bulk. Based on the experimental evaluation, the paper suggests that the technique provides bulk-spam detection that is robust to increased obfuscation efforts by spammers, and low miss-detection of good emails.

Another research by Sarafijanovic et al. [9] proposed an improved open digest algorithm which extends some of the open-digest paper [8] experiments, using the simplest spammer model from that paper. They find that the conclusions of the open-digest paper are rather miss-leading. Then they propose and evaluate, under the same spammer model, a modified version of the original digest technique. The modified version greatly improves the resistance of spam detection against increased obfuscation effort by spammers, while keeping miss-detection of good emails at a similar level.

The modified technique uses the same Nilsimsa hashing function, but instead of producing one digest from the complete email, it produces multiple digests per email, from the strings of fixed length, sampled at random email positions. Basically, they only change the way of producing the digests from emails.

Ying et al. [10] present a new clustering method which is based on the DBSCAN clustering algorithm and Nilsimsa open digest algorithm. Using this method, all emails identified similar artificially are clustered together. The result of the simulation shows that the clustering method based on DBSCAN and open digest performs with higher clustering accuracy than the open digest method but they still suffering from misdetection of some of the spam mails against increased obfuscation effort of spammers which is shorting of open-digest algorithm.

4. PROPOSED ALGORITHM

In this section we will first introduce the current research situation and then explain our proposed algorithm details.

4.1 Research Situation

At present the latest proposed algorithm by Ying [10] used open digest algorithm with DBSCAN clustering algorithm to achieve the highest accuracy of clustering the produced digests of different emails.

Older digest clustering papers used threshold clustering method. This method clusters digests by scanning every digest and comparing each two digests in the digests set. It can ensure that in the final result, each two digests in the same group are similar. In this method, the threshold is determined by experiment. A larger threshold will reduce the similarity in the group and a smaller threshold will increase the number of groups in the result. By analyzing the result, it is easy to find some similar emails are clustered into several groups. Using this method, the results will be different when the input order is different. Clearly, the reason for the problems above is that the shape of the spam digest subspace is irregular.

Using DBSCAN has solved problem related to threshold clustering algorithm but the use of open digest algorithm cause high spam misdetection against increased obfuscation efforts of spammers.

Our proposed Algorithm use the improved digest algorithm proposed by Sarafijanovic [9] with DBSCAN Clustering algorithm. Modified version of open digest greatly improves the resistance of spam detection against increased obfuscation effort by spammers, while keeping miss-detection of good emails at a similar level.

4.2 Improved digest with DBSCAN Algorithm

Based on the problem exists in the threshold spam clustering with open digest algorithm , we propose a new clustering method based on DBSCAN clustering algorithm and improved open digest algorithm. In this section, we introduce the digest generation process first, and then briefly describe the clustering process using the DBSCAN algorithm and the improved open digest algorithm, finally we describe a method for discussing the parameters used by DBSCAN in the anti-spam field.

4.2.1 Digest Generation Process

When a group of mails is received, multiple digests per email are generated. For each mail; it is first trimmed by removing all spaces, and then it is divided into random fixed length strings. The length of each division is 60 characters. For each random fixed length string we generate 265 bit digest using Nilsimsa algorithm.

When all mail are fetched, divided and get digest generated for each division; it is delivered to DBSCAN clustering algorithm to cluster mails based on similarity.

All similar mails (Spam Bulk) are clustered to a single cluster where ham mails are considered as outliers or noise and aren’t included in any spam cluster.

4.2.2 Nilsimsa Digest Space

Nilsimsa digest space is a 256-dimensional space, each dimension values 0 or 1. We define the Nilsimsa digest space as ∂ , define the digest of email as m={s_1, s_2,…., s_n} (n= number of divisions per mail , m $\in$ ∂, $s_i = \{d_1 , d_2 , \ldots\ldots , d_{256}\}$, d $\in \{0,1\}$), define the distance between two digests m1and m2 as follows:

$$distane\ (s_1, s_2) = \sqrt[2]{\sum_1^{256}(d_{1i} - d_{2i})^2} \qquad (Eqn. 1)$$

$$distane\ (m_1, m_2) = {\sum\nolimits_1^3 distance\ (s_x, s_y)}\Big/_{3} \qquad (Eqn. 2)$$

Where x,y are the index of the most three similar divisions between m1 , m2 .

As shown in eqn. 1 and eqn. 2, the digests of divisions of both mails are compared and the average of the distance between most three similar division digests is considered.

Open digest algorithm proposed Nilsimsa Compare Value (NCV). Where NCV between two digests is equal to the number of the equal bits at the same positions in the two digests, minus 128 (for the digests of 256 bits). The higher NCV indicates the higher similarity of the texts from which the digest are computed. The threshold of NCV values proposed to be 74. If NCV is bigger than or equal to 74 then the two mails are similar, else they are different.

In improved open digest space, d_i values only 0 or 1, so the distance is equal to the number of different bits between two division digest. Improved open digest defines that when the distance between two digests is smaller than or equal to 38 (128-90) [9] where 90 is the new NCV Threshold defined by improved open digest, the two digests are similar. If any two digests in the group are similar and the number of digests exceeds the threshold, the group can be called a cluster of spam. The ideal distribution of the spam digests in digest space is as shown in Fig. 1 (a). However, if the spam digests distribute in irregular shapes as shown in Fig. 1 (b), using the threshold clustering method may lead to the result that large cluster is divided into several small clusters, and the number of the clusters increases. Using the DBSCAN can cluster such an irregular shape cluster together into a large cluster.

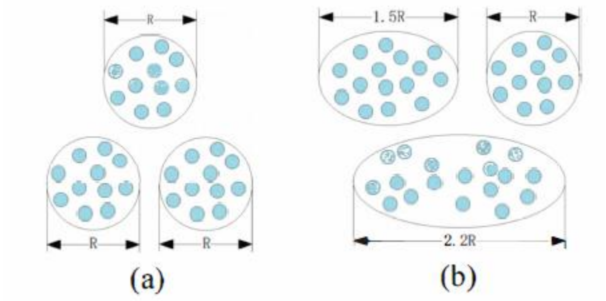

Fig1: distribution of the spam digests in digest space

4.2.3 Clustering mails using DBSCAN

DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a cluster (minPts). It starts with an arbitrary starting point p that has not been visited from the group of points D. This point's ε-neighborhood is retrieved, and if it contains sufficiently many points less than or equal to MinPts it is called a core point and a cluster is started. Otherwise, the point is labeled as noise. Note that p might later be found in a sufficiently sized ε -environment of a different point and hence be made part of a cluster.

If in the range of p's ε radius the number of the elements is less than MiniPts, we can call p as the boundary, p is marked as noise node temporarily. Then, DBSCAN will dispose the next element in set D. As the first and the last step is the same as the threshold clustering method, so the two steps are ignored here. The main workflow of DBSCAN clustering is shown as follows:

Step1: Scan the mail ***p*** in the set ***D*** one by one. Judge whether it has been clustered in a cluster. If so, skip this mail, otherwise turn to Step2. If the scan of all the digests in the set ***D*** is completed, then turn to Step3.

Step2: Get the number of neighbors of ***p*** within the range of ε. This step is done by calculating the distance between ***p*** and all other mails. The calculation of distance between two mails as shown in previous section include the average of the smallest three distances between all of the two mail digests distances. If the number isn't less than MinPts, set the digest ***p*** as the core mail, then scan each of the neighbors of ***p*** and turn to Step l for recursive queries. Finally, all elements from recursive clustering are marked as a new cluster, and then turn to Step l to dispose the next mail of set ***D***.

Step3: Scan all the mails in set ***D***, if a mail isn't in a cluster, it should be marked as a noise mail, and the corresponding email should be regular.

As shown in Fig.2, we set MinPts as 3.There are three mails within A's radius of ε. So it meets the demand, a can serve as a core. Do the recursion from the three digests. Take digest B for example, there are single mail within B's radius of ε. So B is a boundary point. The recursion stops when the boundary digests doesn't meet the density demand.

As there is no other mail within the radius of mail N, the mail N is a noise one, this mail is regular. We should note that all of the mail must be queried. In order to show the process clearly, Fig.2 only shows the query processes of several mails.

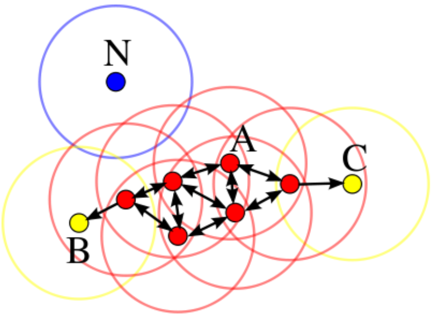

Fig. 2 Large spam mails clustered correctly

5. EXPERIMENTS AND RESULTS

In this section, experiment environments and an evaluation method are introduced. The evaluation process includes the accuracy of proposed algorithm based on produced clusters of spam mails.

5.1 DBSCAN Parameters

DBSCAN requires two parameters: ε (eps) and the minimum number of points required to form a cluster (minPts), for the first parameter ε (eps) value based on improved open digest algorithm it will equal to 38 since the proposed NCV equals to 90, so the threshold of distance between two mails will be 128 minus 90 which equals to 38.

For the second parameter we have experiment many values of minPts against accuracy where the accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantity's actual (true) value. So we can define accuracy as following:

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{Number\ of\ spam\ mails\ Clustered}}{\boldsymbol{number\ of\ ALL\ Spam\ Mails}} \qquad \boldsymbol{Eqn}.1$$

To estimate the best value of MinPts for DBSCAN we have used a collection of spam groups from Spam assassin public corpus [12]. The experiment of calculating the best minPts that provides highest accuracy includes the following steps.

Group of mails are sampled randomly from the used spam repositories including 27 mails from 20021010_spam repository, 55 mails from 20030228_spam repository and 37 mails from 20030228_spam_2 repository from the Spam assassin public corpus [12]);

Then proposed algorithm tested with different values of minPts parameter including 2 – 7 range. The results as shown in figure 3

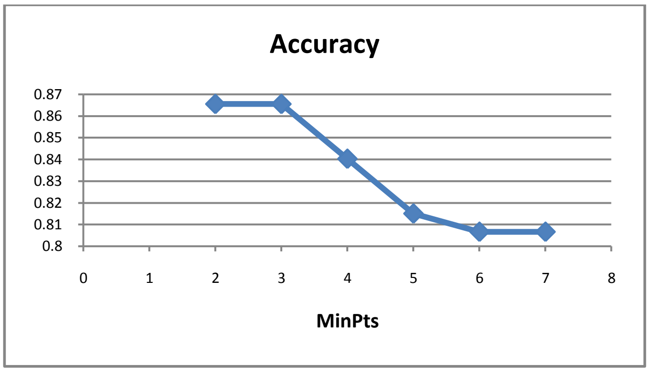

Fig. 3 Best MinPts Value

Based on the results shows in Fig. 7 for the range of minPts from 2 to 7 MinPts=3 provides the highest accuracy so we have used minPts=3.

5.2 Clustering Evaluation method

The accuracy of clustering is determined by comparing the clusters obtained by the experiments with the real spam clusters specified by Spam assassin public corpus [12] using accuracy as mentioned in equation 3.

Precision and recall are the basic measures used in evaluating search strategy. Precision is the ratio of the number of relevant records retrieved to the total number of irrelevant and relevant records retrieved. It is usually expressed as a percentage where recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database. It is usually expressed as a percentage.

$$Precision = \frac{T_p}{T_p + F_p} \qquad Eqn.5$$

$$Recall = \frac{T_p}{T_p + F_n} \qquad Eqn.6$$

Where T_p is the number true positives (spam mails that has correctly clustered), F_p is the number of false positive (ham mails that clustered as with spam mail) and F_n is the number of false negatives (Spam mail not clustered and specified as ham mail).

5.3 Evaluation Experiments and Results

To check the accuracy of the proposed algorithm against latest algorithms using accuracy equations mentioned above. The proposed algorithm is experimented against latest algorithms including threshold clustering with open digest algorithm, threshold clustering with improved digest and DBSCAN with open digest algorithm Steps of the experiment include:

- 90 of mails are sampled randomly from the used spam repository (we use 20030228_spam.tar.bz2 spam repository from the Spam assassin public corpus [12]);

- For the measured algorithms parameter the threshold value for open digest is 54 where higher values indicates similar mails. The threshold value for improved digest algorithm is 90, if two mails have similar bits is larger than 90 they are considered to be similar .For both previous algorithm the minimum number of points in a single cluster is 4. For DBSCAN with open digest we assign the following parameters MinPts= 3 and ε (eps) =128-54=74 where 54 is the threshold value for open digest .The proposed algorithm Minpts=3 and ε (eps) =128-90=38 where 90 is the threshold of the improved digest.

- The results are shown in figure 4

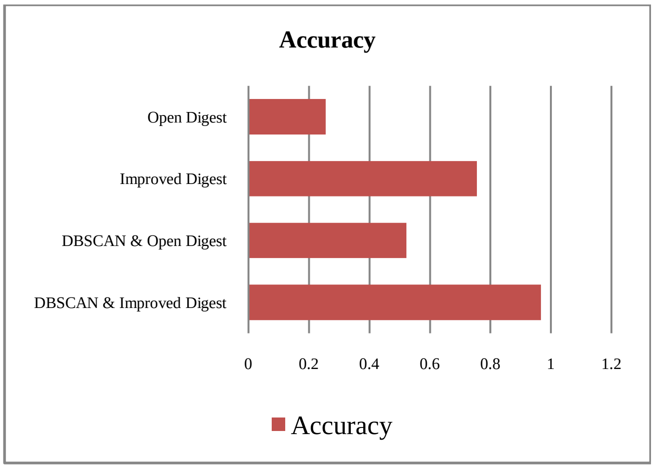

Fig. 4 proposed algorithm accuracy against latest algorithms for 90 spam mails of 20030228_spam group

Based on results shown in figure 4 DBSCAN with improved digest provided the highest accuracy where it can capture 87 spam emails out of 90 spam mails from 20030228_spam group and 197 out of 200 spam mails from 20021010_spam group , as we can see from the result the impact of using DBSCAN with the improved digest algorithms it has raised the accuracy of improved digest only to about 30%. The proposed algorithm has exceeded the accuracy of DBSCAN with open digest with about 30%.

The third experiment examine the precision and recall values, experiment includes the following steps:

- 60 of mails are sampled randomly from the used spam repository (20030228_spam.tar.bz2 spam repository from the Spam assassin public corpus [12] is used);

- 20 of mails are sampled randomly from the used spam repository (20030228_easy_ham.tar.bz2 ham repository and 20021010_hard_ham.tar.bz2 from the Spam assassin public corpus [12] is used. The hard ham group contains spam messages which are closer in many respects to typical spam.

- The four algorithms have been experiment against the specified number of spam mails and ham mails, recall and precision has been calculated based on equation 4 and equation 5. The results is shown in figure 5

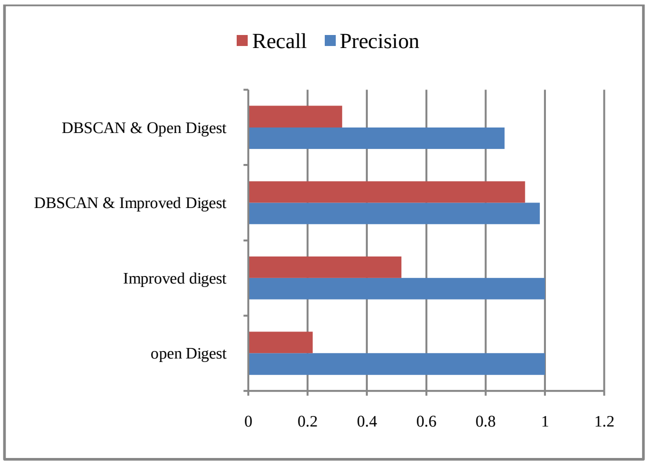

Fig. 5 Precision and recall values of proposed algorithm against recent Algorithms

Based on results shown in Figure 9 the proposed algorithm provides the best recall and precision values. As we can see the recall value for the proposed algorithm is the highest value since most of spam mails have been clustered but other algorithms don’t cluster high number of spam and consider it as ham mails.

5. CONCLUSION AND FUTURE WORK

In this paper, a new algorithm for spam detection is proposed using the DBSCAN clustering algorithm and the improved open digest algorithm to cluster the emails and identify the spam. By simulation this algorithm performs better than the threshold clustering algorithm and DBSCAN with open digest algorithm. The using of improved open digest with DBSCAN has improved the accuracy, precision and recall values of spam searching.

Based on the improved open digest algorithm, further research to optimize the length of the fix size division of a single mail.

6. REFERENCES

[1] C. Pu and S. Webb, 2006. Observed trends in spam construction techniques: A case study of spam evolution. In Proc. of the 3rd Conf. on EMail and Anti-Spam.

[2] L.F. Cranor and B.A. LaMacchia, 1998. Spam! Communications of the ACM.

[3] Wikipedia , [online], http://en.wikipedia.org/wiki/SpamAssassin

[4] Rhyolite distributed checksum clearinghouse. http://www.rhyolite.com/dcc/

[5] Jesse Kornblum, 2006, "Identifying almost identical files using context triggered piecewise hashing", Digital Investigation, vol. 3(sl):9 1-97.

[6] Zhang Jianzhong, Lu Hongbo, Lan Xiaofeng, Dong Dafan, 2008, "DHTnil: An approach to publish and lookup Nilsimsa digests in DHT". Proc. of the 2008 International Conference on High Performance Computing and Communications (HPCC-08), Dalian, China.

[7] Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaowei Xu,1996,” density-based spatial clustering of applications with noise - DBSCAN”.

[8] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, 2004,”An Open Digest-based Technique for Spam Detection “ , International Workshop on Security in Parallel and Distributed Systems

[9] Slavisa Sarafijanovic, Sabrina Perez, JeanYves Le Boudec, 2008, " Improving Digest Based Collaborative Spam Detection," MIT Spam Conference.

[10] Wu Ying, Yang Kai, Zhang Jianzhong, 2010, " Using DBSCAN Clustering Algorithm in Spam Identifying ", 2nd International Conforence on Education Technology and Computer (ICETC).

[11] J. Han and M. Kamber, 2001, “ Data Mining: Concepts and Techniuqes”. Morgan Kaufmann Publishers, SanFrancisco, CA,

[12] SpamAssassin-Public-Corpus. http://spamassassin.org/publiccorpus/, March 2013.