Journal of Kerbala University , Vol. 12 No.2 Scientific . 2014

SMS Spam Identification Based on Message Duplication Detection by Cuckoo Filters

تحديد الرسائل القصيرة غير المرغوبة بناء على اكتشاف تكرار الرسائل باستعمال مرشح الوقواق

Saif Ali
Computer Science Department-Kerbala University
Saif.a@uokerbala.edu.iq

ABSTRACT

Short message service (SMS) spamming has become a large problem due to the wide spread of smart phones in the past few years. Modern smart phones processing power and extended connectivity has been employed by new spamming techniques to send spam messages from multiple infected devices controlled by command centres. Besides the annoyance for the receiver, this new breed of SMS spamming is causing financial loss for the infected devices owners since these devices has been used as tools for SMS transmission and spending the owner credit in the process. These new methods introduce a challenge for the telecommunication companies since it requires new techniques to identify and stop spam messages and the suggested method provide one practical solution for this problem. This paper presents a method to detect spam messages using the chronological characteristics of the spamming campaigns and the similarity among the spam messages' contents which servers a single goal. A system has been built to recognize the repeated transmission of identical or near identical spam using the compact and high-performance Cuckoo filter. Real SMS messages were used to evaluate the system performance and detection rate.

Keywords: SMS, n-gram, Cuckoo filter, Bloom filter, spam, Bayesian

المستخلص –

أصبحت الرسائل النصية القصيرة غير المرغوبة المستلمة مشكلة كبيرة بسبب الانتشار الواسع لأجهزة الهواتف الذكية في السنوات القليلة الأخيرة. ان ما تمتلكه الهواتف الذكية الحديثة من امكانيات المعالجة المتطورة وسهولة الاتصال قد تم توظيفه من قبل عمليات ارسال الرسائل غير المرغوبة عن طريق عدد من الأجهزة المصابة والمسيطر عليها من قبل مراكز السيطرة. بالإضافة للإزعاج الذي تمثله هذه الرسائل غير المرغوبة للمستلم فأن هذا الأسلوب الجديد في الارسال يسبب خسائر مادية لمالكي الأجهزة المصابة التي يتم استخدامها في الارسال والتي يتم استخدامها كأدوات لأرسال الرسائل غير المرغوبة وصرف رصيد المالك خلال هذه العملية. تمثل هذه الطرق الجديدة تحديا لشركات الاتصالات لما تتطلبه من حلول جديدة لتحديد وإيقاف هذه الرسائل ويقدم البحث احدى الطرق فعالة لمعالجة المشكلة.

يقدم البحث طريقة عملية لاكتشاف الرسائل غير المرغوبة عن طريق الخواص الايقاعية والموقوتة لحملات ارسال الرسائل غير المرغوبة التي يجمعها هدف واحد. لقد تم تصميم وبناء نظام يقوم بالكشف عن التشابه في محتويات الرسائل القصيرة المتطابقة والمتشابهة باستعمال هيكل البيانات الوقواق ذو الأداء العالي. تم استعمال رسائل نصية حقيقية لتقييم أداء النظام ونسبة الاكتشاف في بيئة مطابقة للواقع.

I. INTRODUCTION

Short messaging service (SMS) has become one of the most widely used communication mediums after the fiery penetration of mobile phones. The low-cost and the ability to send multiple messages to several destinations at a relatively short period are the top reasons behind the service popularity in modern communities. However, the same characteristics have been used by spammers to target the phone users by sending various kinds of annoying messages. The goals of these unrequested messages are normally advertising, though, SMS spamming might be used as a part of more sophisticated scams like urging the receiver to call or send SMS to pre-configured high-cost destinations. Formally, SMS spam is any unsolicited message delivered to a mobile phone through

text messaging[1].The numbers of this kind of messages experience a yearly growth larger than 500%[2] which made SMS spamming a very real worldwide communication threat.The wide spread of SMS spam negatively affects the customer satisfaction and the entire SMS confidence in some countries. For example,SMS spam in some parts of Asia reaches 30% of a total number of messages[3].SMS spamming is very similar to email spamming in the goals and even the words used in the message or the email body.Based on this basic similarity,content-based approaches in email spam detection have been usually employed to detect SMS spam and spammers[4].However, content-based algorithms used in email spam detection are less capable in SMS spam detection because of the short length of the SMS message[2].In addition to the senders'blacklist and white list filters,keywords filtering was one of the earliest methods to filter SMS spam. Bayesian algorithm has been adopted by Zhang et al. in [5] to provide more accurate spam identification filters.

This paper is organized as follows; section II will briefly list some researches which share similar methodologies to the one presented in this paper to identify spam SMS messages. Section III will provide definitions and some background about the terms and data structures used throughout the paper. In Section IV, the mathematical model of the proposed method will be introduced. In section V the experiments will be demonstrated and then discussed in section VI. The last two sections will highlight the future work and the conclusions.

II. RELATED WORK

In [6] Coskun and Giura presented an approach to recognise SMS spam based on the temporal behaviour of the spam and the SMS content. The method stated that each SMS would be divided into blocks then add these blocks to a counted Bloom filter to examine counters of message contents. The counters will be examined for any unusually high number of near-duplicated messages over a short period. The authors used a shingling variation to divide the messages into blocks named n-grams. N-grams of a text message are all the sub string of size n for that message which can be found by getting the first substring of size n, shift one character from the start and get n characters substring. Shifting operations will continue until reaching the end of the message. Some tests have been done to calculate the best value for the block size and it has been found to be 5.

In [7], the researchers presented a content based spam filtering based on Bayesian classifier with word grouping and test their wok against real SMS messages. Using word grouping, it was possible to reach high accuracy of spam identification. However, this method (and similar Bayesian based methods) force the operator to collect a list of know spam messages and actively maintain this list to get accurate spam identification.

III. BACKGROUND

III.1. Definition and Characteristics of SMS Spamming

Unlike legal SMS messages, or simply hams, Spam SMS refers to those sent in bulk with illegal or violating content or those received violating subjective preference of mobile phone users and causing harassment to users objectively[5]. Like email spamming, SMS spamming is based upon the principle of sending high messages volumes to multiple destinations at a high sending speed and low cost per message hopping that some receptions will actually do what the SMS message urges the receiver to do like buying a product or joining a service.

The spammer will have to transmit many identical or near identical messages in a short time to achieve the campaign goals. Near identical messages are the messages generated by the spammer after changing one or more characters in every transmitted message. The change might be a combination of addition, deletion or replacing a character or more in the original spam message template. Adopting this method will produce multiple unique messages with the same basic meaning and practically shield the messages from the simple message duplication tests. This method can be used to an extent since only few characters can be changed without turning the message meaningless or suspicious, consequently, missing the spammer goal.

III.2. Blacklist and Whitelist SMS Spam Filtering

Usually, spammers can't change the sender phone number while using normal prepaid phone accounts to send SMS. Based on this hypothesis the blacklist and whitelist SMS Spam filtering methods has been built. Black and white phone number lists consist of known spam senders and trustful SMS senders respectively. The blacklist approach will enable anyone to send the SMS except the tagged spammers. At the SMS arrival to the system, the sender phone number will be checked in the blacklist and the SMS will be dropped on positive matches. The whitelist method on the other hand is inclusive, which is mainly used for confirming legal SMS source to reduce exclusion error of the blacklist[5]. Blacklist and whitelist filtering are considered to be very efficient and none resource consuming due to the methods' simplicity. Still, these methods requires the operator to categorize phone numbers manually. Furthermore, legal SMS messages sent from blacklisted phone number will be discarded which is much more offensive than sending spam messages, for example, a shopping site might send a spam message offering new item, and a legitimate message containing phone number authentication code[8].

The counts of smart phones which are powered by modern operating systems like Android or the iPhone Operating System (IOS) has been increased dramatically. But these smart phones can be easily infected by malwares when used with incaution. A survey shows the amount of malware identified on the Android platform has increased about 472% during the period June 2011 to November 2011[9]. These infected devices might acts together as botnets controlled by the spammer who orders the botnet to send SMS spam, therefore, using the blacklist or whitelist methods will identify many legal but infected phones as spammers and blocking ham messages from these phones as a result.

III.3. Bayesian Spam Filtering

Bayesian spam filtering statistically classify SMS as being spam or ham by using the probabilities of finding some predefined words inside the message text. The process requires predefined training message lists, a blacklist contain spam message samples, and whitelists contain definitely legitimate message samples. The spam filter will be trained to learn the probability of finding each word in spam or ham messages by use of both black and white lists. Whenever a new SMS arrives, the filter will extract all the words and compute a score with the help of the training lists. This score is then compared with a threshold parameter to decide on an SMS is spam or ham[10]. The main problem of this method is the short length of the SMS. SMS message can only be 160 characters in length when using the default encoding and 70 characters when using the 2-byte Universal Character Set (UCS2) encoding [11]. Bayesian spam filtering should take into consideration this problem of fewer words to examine and thus less information available to identify the SMS message class, whether the message is a spam or a ham[12].

III.4. Cuckoo Hashing and Cuckoo Filter

The basic Cuckoo hashing has been described by Pagh and Rodler in [13]; a high performance dictionary data structure which can be used for any value matching tasks. It is used to search for a previously seen data with a constant worst case lookup time of 2 read operations. The dictionary uses two hash tables, T1 and T2, each of length r, and two hash functions $h_1, h_2 : U \rightarrow \{0, \dots, r-1\}$. Every key x is stored in cell $h_1(x)$ of T1 or $h_2(x)$ of T2, but never in both. To lookup a value y, only the cells addressed by $h_1(y)$ in T1 and $h_2(y)$ in T2 will be accessed to be tested against y and return a positive lookup if at least one of the cell does contain y [13]. While inserting a new key in the dictionary, cell $h_1(x)$ of T1 will be checked to determine if it is occupied. If not, then it will be set to contain x. If the cell was already used then it will be set x anyway but the old content of that cell will be inserted in to T2 using the same procedure, and so forth iteratively.

Fan et al. presented partial-key cuckoo hashing in [14] .The authors used a modified version of Cuckoo hash to build a dictionary data structure. The modified Cuckoo filter stores a fingerprint (a hash) of the value instead of the actual value in one table instead of two. Also the hash functions

h_1 and h_2 are no longer independent of each other. For an item x, the hashing scheme calculates the indexes of the two candidate cells i_1 and i_2 using Eq.(1) [14]:

$i_1 = HASH(x)$ (1)

$i_2 = i_1 \oplus HASH(x's\ fingerprint)$ (2)

The exclusive-or operation in Eq. (2) ensures that i_1 can be calculated using the same formula from i_2 and the fingerprint.

In summary; inserting a value (x) in the filter will be done by storing its fingerprint in the location i_{1x} or i_{2x} (the store locations i_1 and i_2 for the value x) of the table using Eq.(2). If none of them is empty then one of these locations will be overwritten. Its original contents (y's fingerprint) will be displaced to the alternate location(i_{2y})which is the store location, i_2,for the already inserted value y . Searching the filter for the value x will return true if i_{1x} or i_{2x} actually holds x's fingerprint.

IV. DETECTION METHOD

This section will present the formalization of the proposed detection method used to name spam messages by targeting the temporal behaviour of the spamming campaigns. The method is based upon the observation that spammers typically use one device or more to send multiple identical or near duplicated messages during a short period to achieve the campaign's goals.

IV.1. Detection of Identical or near identical Messages

To find identical or substantially alike messages the SMS text will be represented by a set of blocks or sub-strings. Any two messages will be considered to be identical if they have the exact same set of blocks. They will be near identical or similar messages if there are few unmatched blocks between them, otherwise they are said to be unrelated messages. The generation of the message blocks will be done using a shingling variation similar to the one described in [6] where each message will generate N n-grams. If the message length is *S* and the n-gram length is *L* then N can be calculated using Eq.(3) [6]:

$N = S - L + 1$ (3)

Increasing the number of unique n-grams per message will enhance the result of finding similar or identical messages.

The proposed method follow a similar path to the work presented in [6] with two main differences, the filter type and the n-grams generation method. The proposed method use Cuckoo filters instead of Bloom filters since they have a better performance and space efficacy as presented in [14]. The second enhancement introduces a method to produce more n-grams for the same message and thus increasing the identification accuracy.

A trailer consisted of a special character to mark the original end of the message followed by the first $L-1$ characters of the message will be appended to the message. This addition will generate L additional unique n-grams per message. Substituting S by $S + 1 + (L - 1)$ in Eq. (3) will produce:

$N = S + 1$ (4)

For instance if the message is "$flamingo$" and the value of *L* is 3 then the original set of the n-grams will be {"fla","lam","ami","min","ing","ngo"}. By appending Ω to represent the original position where the message ended followed by the first two characters which are "fl". The new message will be "$flamingo\Omega fl$" and the related n-grams set{"fla","lam","ami","min","ing","ngo","$go\Omega$","$o\Omega f$","Ωfl"}. The block "$go\Omega$" in the previous example is unique to any message ends with a "go". The next block , "$o\Omega f$" will appear only in the messages that ends with an "o" and starts with an "f". Similarly the block "Ωfl" is uniqe to the messages which starts with an "fl".

As shown above, these additional 3 n-grams provided additional identification information which were unseen before adding the extra trailer.

Message x_1 will be tested for similarity with message x_2 based on Jaccard similarity coefficient using an approach similar to the one presented in [6] with the adoption of the new equation Eq.(4) to calculate the n-grams total number of n-grams. if the ratio of the matched n-grams in x_1 to the

total number of n-grams of x_1 is larger than θ , therefore, Eq.(5) can be used to identify similar messages by assuming that only β n-grams from x_1 dose not appear in x_2 n-grams and calculating the number of n-grams using Eq.(4):

$$\theta < 1 - \frac{\beta}{S+1} \quad (5)$$

Changing one character in any message will alter L n-grams which are the blocks includes that character, therefore, changing d characters will change $d.L$ n-grams if there are no altered characters closer than L . However, changing d characters will alter less than $d.L$ if the changes are closer than L to each other. To neutralize the effect of changing d characters, β in Eq.(5) will be substituted by $d.L$ as shown in Eq.(6):

$$\theta < 1 - \frac{d.L}{S+1} \quad (6)$$

IV.2. Spam Identification

The SMS will be tagged as a spam if identical or near identical messages appear more than ρ times in the period t , where ρ is a predefined threshold of the maximum number of identical or near identical messages per the period t. t will be divided equally into n segments $t_1, t_2, \ldots t_n$ and for each segment t_i the n-grams of all the messages transmitted during t_i will be denoted by $M_i = \{m_{i1}, m_{i2}, \ldots., m_{i3}\}$. After running the system for $t + (t/n)$ seconds, the n-gram set of the oldest segment, M_1 , will be discarded and the new set M_{n+1} will be created.

The n-grams of any new message will be looked up for similarities in all the sets, M_2 to M_n ,using Eq. (5) and the message will be marked as a spam if the message appears more than ρ times. n Cuckoo filters were used to represent the sets M_1 to M_n and each n-gram of each new message will be looked up in all the filters upon arrival then added to the current Cuckoo filter. Using the filters will enable real time lookup operations to be implemented efficiently computational and space wise.

The proposed method has been based on the assumption that duplicated messages would not be added to the same filter twice. The reason for this assumption is that Cuckoo filters does not provide counters or any mechanism to save the number of identical additions to the filter, so, if the a message has been found to in the current filter M_1 then it should be marked as spam instantly.

The selection of ρ and t will define the maximum acceptable rate of message duplication formulated in Eq.(7) :

$$Maximum\ allowed\ duplication\ rate\ =\ \rho\ /\ t \quad (7)$$

The number of the Cuckoo filters, n , should be always larger than or equal to ρ , however, every additional filter will cost additional two memory read operations at most while checking each n-gram. On the other hand, using fewer Cuckoo filters will increase the number of messages per Cuckoo filter and thus increase the number of false positives as well as increasing the chance of having two or more duplicated messages while generating the current Cuckoo filter which is not permitted by design.

V. EXPERIMENTS

V.1. Dataset

The dataset used in this work for the tests and experiments was provided by Omnnea wireless telecom, an Iraqi mobile operator. Arabic SMS message has been collected and prepared by the operator by hashing the source and destination fields (using MD5 hash function) to minimize the privacy concerns. Using Arabic data required dealing with the shorter SMS text (and fewer n-grams as a result) since Arabic SMS use UCS2. However, the detection method is the same.

V.2. N-gram size selection

The worst case scenario for Eq.(6) will occur while processing the shortest message submitted to the system. The shortest messages has been selected to be 30 characters through all the experiments and tests. This number was selected since only 70 Arabic characters per SMS are allowed and it is

unlikely that the spammer will be able to use messages less than 30 character in length to gain effective results.

The n-gram length, L , has a crucial effect on the accuracy of the system. Choosing a low value will cause many false positives but it will increase the system robustness against the minor messages changes, conversely, choosing a large value will reduce the system sensitivity against message modifications. An experiment has been designed and implemented to highlight the best value for the n-gram size which practically reduce false positives and guarantee acceptable near duplicate detection. 36368 unique messages has been divided equally into two groups, the first group was used to fill a Cuckoo filter and the second one was tested using Eq.(5) against that filter.

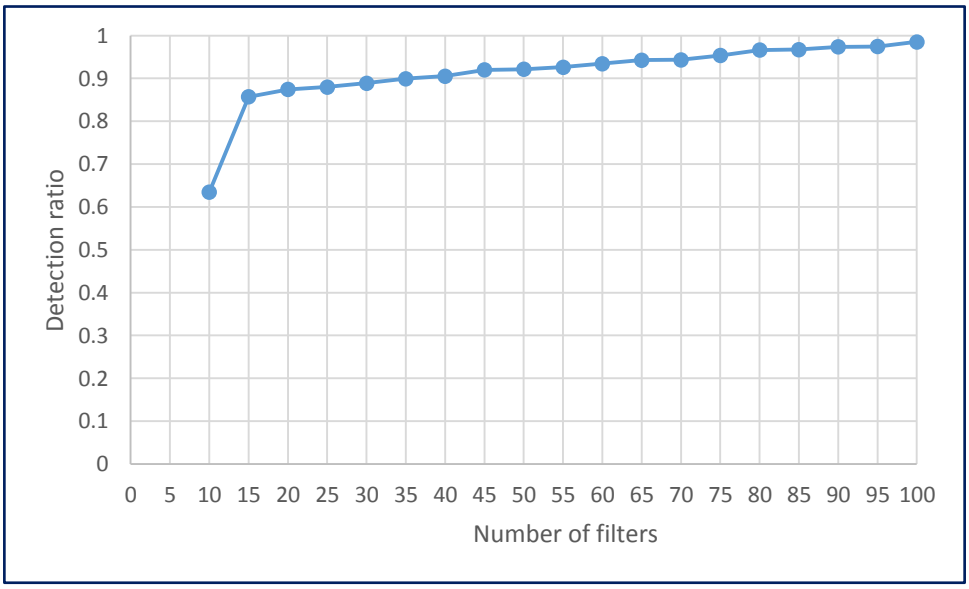
Through the experiment, the value of L has been changed from 2 to 30 in a step of one, the value of θ has been calculated accordingly using Eq. (6) to guarantee the detection of changing a single character. All the messages in the experiment were known to be ham and unique. Any degree of similarity between messages n-grams of the two groups was caused randomly by the users so it might occur in run time causing false positives. The experiment results were examined to select the smallest value for L which produced the lowest duplication detection rate. Table 1 shows the variation of the false positive of near duplicate messages against different n-gram size L. As shown in Table 1, the n-gram size of 9 produced the smallest number of false positive and having a larger value did not provide any enhancements so the n-gram size of 9 has been used during all the following tests.

Table 1n-gram size effect on the false positive rates

L	false positive rate
2	0.35795205
3	0.17218434
4	0.05125385
5	0.00516938
6	0.00098988
7	0.00032996
8	0.00021997
9	0.00005499
10	0.00005499
11	0.00005499
12	0.00005499
13	0.00005499
14	0.00005499
15	0.00005499
16	0.00005499
17	0.00005499
18	0.00005499
19	0.00005499
20	0.00005499
21	0.00005499
22	0.00005499
23	0.00005499
24	0.00005499
25	0.00005499
26	0.00005499
27	0.00005499
28	0.00005499
29	0.00005499
30	0.00005499

V.3. Number of Cuckoo filters

An experiment was conducted to evaluate the effect of changing the number of Cuckoo filters on the detection ratio of similar messages. The experiment has been designed to simulate a short message centre receiving constant rate of 25 messages per second. Any message that appears more than 10 times during 5 minutes will be considered as a spam. The parameters ρ and t have been set to reflect the simulated system. ρ has been set to 10 and t was set to 5 minutes. 200000 messages, containing 1450 known spam messages, has been scanned to identify spam messages using the proposed method. The experiment was conducted 20 times using different number of Cuckoo filters, n, starting with 10 and adding 5 Cuckoo filters at each step. 100 Cuckoo filters were used in the last step where the detection rate was 0.9855 and adding more filters was practically unnecessary.
Figure(1) shows the detection rate variation against the number of filters.



Figure(1) Detection rate for different number of Cuckoo filters

VI. Results and Discussion

As shown in Figure(1), increasing the number of filter will increase the detection rate. This result was expected since each filter represented a period of time in which all the transmitted messages will be grouped together. The Cuckoo filter abbreviated multiple identical messages into one, consequently, it caused the system to miss calculate multiple transmission of the identical message in the single Cuckoo filter and always return one match only. Using more Cuckoo filters reduced the effect of the problem because fewer number of messages will be grouped in a single filter. The new Cuckoo filters added match count for the same messages. Using more Cuckoo filters had a performance drawback since they required extra processing and memory access to check every message for matches in the additional Cuckoo filters.
Compared to the method described by [6] , the proposed method has the below advantages:

1- Does not require a training set which is used to calculate the thresholds of the counted Bloom filters bins.
2- More unique n-grams will be generated for the same SMS, consequently, less false positives.

VII. Future work

The effectiveness of the proposed method depends on the number of filters used and the targeted messages duplication rate. To target high speed spammers, only a small number of messages should be grouped in the single filter to avoid having multiple matches in the same filter. This will enhance spam detection rate for the spam messages which are transmitted with relatively low delay between each other or simultaneously transmitted using multiple botnet controlled

devices. On the other hand, identifying slow spammers require different setup in which the system should store the messages' n-gram hashes for an extended periods. As a result, small number of large filters should be adopted.

It is possible to build a system consisting of two independent spam identification stages. The first stage will be used to identify high speed spammers while the next stage setup will targets low rate spammers.

Another issue with the proposed method is the fact that mobile users tend to send very similar greeting massages in the national or religious holidays. These messages will be tagged as spam since they appear frequently in the platform during a short period. One possible solutions is using a whitelist to skip all the messages which contains words related to the occasions.

VIII. CONCLUSIONS

The proposed method introduced practical method to identify spam messages based on the temporal behaviour of the spam messages transmission and the identical or near identical spam messages content during a single campaign. The main design was based on the efficient Cuckoo filter and the careful fragmentation of the SMS text to discover the similarities between different messages.

The method has been tested against real life SMS messages and the results showed a very low false positives rate (0.00005499 for the n-gram size of 9) and high spam detection rate (more than 0.90 when using 40 Cuckoo filters and up to 0.985 for 100 Cuckoo filters setup), thus, the method can be practically used as a complete spam detection solution or as a pre-processing stage for a more complex and computationally expensive detection methods.

REFERENCES

[1] K. Yadav, P. Kumaraguru, A. Goyal, A. Gupta, and V. Naik, "SMSAssassin: Crowdsourcing driven mobile-based system for SMS spam filtering," in *Proceedings of the 12th Workshop on Mobile Computing Systems and Applications*, 2011, pp. 1–6.

[2] I. Murynets and R. P. Jover, "Analysis of SMS Spam in Mobility Networks," *International Journal of Advanced Computer Science*, vol. 3, no. 7, 2013.

[3] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: new collection and results," in *Proceedings of the 11th ACM symposium on Document engineering*, 2011, pp. 259–262.

[4] Q. Xu, E. W. Xiang, Q. Yang, J. Du, and J. Zhong, "SMS Spam Detection Using Noncontent Features," *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 44–51, Nov. 2012.

[5] H. Zhang and W. Wang, "Application of Bayesian Method to Spam SMS Filtering," in *Information Engineering and Computer Science, 2009. ICIECS 2009. International Conference on*, 2009, pp. 1–3.

[6] B.Coskun and P.Giura,"Mitigating SMS spam by online detection of repetitive near-duplicate messages,"in *Communications(ICC),2012IEEE International Conference on*,2012,pp.999–1004.

[7] D. Belem and F. Duarte-Figueiredo, "Content filtering for SMS systems based on Bayesian classifier and word grouping," in *Network Operations and Management Symposium (LANOMS), 2011 7th Latin American*, 2011, pp. 1–7.

[8] A.K.Uysal,S.Gunal,S.Ergin,and E.S.Gunal,"A novel framework for SMS spam filtering," in *Innovations in Intelligent Systems and Applications (INISTA), 2012 International Symposium on*, 2012, pp. 1–4.

[9] R.Ramachandran,T.Oh,and B. Stackpole, "Android Anti-virus Analysis," *ASIA and SKM*, 2012.

[10] C. Khemapatapan, "Thai-English spam SMS filtering," in *Communications (APCC), 2010 16th Asia-Pacific Conference on*, 2010, pp. 226–230.

[11]N. J. Croft and M. S. Olivier, "A silent SMS denial of service (DoS) attack," *TechRepublic White Paper*, 2007.

[12] lckin Vural and H.S Venter, "Combating Spamming Mobile Botnets through Bayesian Spam Filtering," in *"The Internet of Things - Smart Homes & Cities"... Evolution or Tsunami*, 2012.

[13] R. Pagh and F. F. Rodler, "Cuckoo hashing," *Algorithms — ESA 2001. Lecture Notes in Computer Science*, vol. 2161, pp. 121–133, 2001.

[14] B. Fan, D. G. Andersen, and M. Kaminsky, "The Cuckoo Filter: It's Better Than Bloom," presented at the Networked Systems Design and Implementation, 2013.