Filtering SMS Spam and Implemetion in Android

Sneha Dalvi
University of Georgia
Athens, Georgia, USA
Email: sneha.dalvi25@uga.edu

Dr. Roberto Perdisci
University of Georgia
Athens, Georgia, USA
Email:

***Abstract*—SMS spam are unsolicited, unwanted text messages sent on mobile phones and include commercial advertisements. These messages annoy mobile phone users and also impose risk of financial fraud or malware downloads. In this research project, we attempt to find a technique to filter spam SMSs by studying spam features and the way in which spammers send spam messages. So we filter spam messages based on the length of sender number and content of the SMS. We also check if a message contains spam features such as spam keywords, some special characters or URLs. To check if an SMS contains spam keywords, we first create a list of common spam keywords and assign weight to each of them according to their frequency. We sum up the weight s as we find spam keywords and other spam features and finally compare the sum with a defined threshold. If it is beyond the threshold, then the SMS is classified as spam. We implement these techniques in an Android application, named as SMSSpamFilter. We then evaluate the schemes accuracy with a dataset of 5,574 messages. Finally, we discuss failure cases, ways to minimize false positives and ways to improve accuracy.**

I. Introduction

Like email spam, Short Message Service (SMS) spam or mobile spam is a widespread issue as there are billions of mobile phone users. SMS spam are nothing but unsolicited, unwanted text messages sent on mobile phones. Most of the SMS spam include commercial advertisements and promotions of service plans. Such SMSs distract user, waste their time in reading and deleting them, flood their inbox. They are particularly annoying for the recipients who are charged for incoming messages. Some spam messages are sent with the intent of hacking or malware attack which imposes higher risk to smartphone users as smartphone connects to internet and contains personal information. So, there is risk of financial fraud and malware downloads. A lot of research has been done to provide anti-spam solution to deal with this problem. Existing solutions suggest different techniques to filter SMS spam such as white and black listing, content based filtering with Bayesian filters and other techniques. But some of them were not evaluated on real system.

In this research project, we attempt to find a technique to filter SMS spam and implement this technique in an Android application. We studied different ways in which spammers send spam SMSs in order to study spam features. Then, we studied SMS content for spam features and used content based filtering technique. Spam messages generally contain words like free, win, prize, etc. They also contain unusual distribution of punctuation marks for e.g. BUY!!!! or percentage character indicating discounts or dollar character indicating rates. Many of them contain URLs. Thus, all of these spam features are extracted from an SMS to determine whether it is a spam message or not. Then, the proposed framework is implemented in an Android application. This application captures incoming SMS and determines whether the SMS is spam or not. If it is a spam, the SMS is silently forwarded to a different folder instead of inbox and the mobile user is not notified. Whereas, if it is a legitimate SMS, it is saved to inbox and the user is notified as normal.

We evaluate the systems performance and accuracy with a collection 5,574 messages which include spam as well as legitimate messages. It is found that the system detects spam message with 60% accuracy. We then discuss reasons where the application fails to detect spam SMS. We suggest techniques that can be used to improve accuracy and minimize false positives. We also discuss better technique different than the proposed technique.

December 11, 2014

II. Methodology

There are different ways in which spammers can send spam messages as these ways allow people to send SMSs with no cost, in an anonymous way or in bulk number. First, there are various free anonymous text messaging websites such as TxtEmNow.com, Sendanonymoussms.com, etc. But, these sites are not reliable. Second, we found an interesting way to send free SMS via email. If you know carrier of a mobile number then you can write email address as [10-digit-phone-number]@[carrier-texting-email-domain] and this mail will be sent to that 10-digit mobile number. For example, 1234567890@txt.att.net in case AT&T carrier. Similarly other SMS gateway listings are: Verizon: @vtext.com, TMobile: @tmomail.net, Pinger: @mobile.pinger.com. So, spammers use script to spam out messages to random phone numbers. We observed that, the SMS received by this technique had 9-digit sender number (e.g. 1-210-100-005) instead of normal 10-digit number, in case of AT&T carrier. Third, there exists services that allow to send bulk SMSs for marketing purposes for e.g. Twilio. Such services generally provide 5 or 6 digit short codes to send bulk SMS. Fourth, services such as Google Voice, Yahoo Messenger also allow us to send free SMS. For this, you need to have the respective accounts. Google Voice provides a 10-digit number to send SMS. Whereas the SMSs which are sent from Yahoo Messenger had 9-digit sender number such as 1111-440-601 and the number increments by one for next SMS and so on. Fifth, there are some spam SMSs which had sender as an email address instead of a number. Thus, we concluded that if an SMS does not have a normal 10-digit sender, then it is possibly a spam message.

Then, to implement content based filtering, we studied sample spam messages to collect spam keywords. We collected 130 spam keywords and assigned a weight to each of them depending on their frequency and likelihood in spam messages. A spam keyword will have higher weight if it has greater frequency. So, an incoming SMS is checked for these spam keywords and if a keyword is present in the SMS, its weight is added to the SMS_weight. We also checked if a word in SMS contains currency character ($) which refers to money, or a percentage sign (%) which refers to discount, or loner punctuation marks (!!!!), then add some weight to the final SMS_weight. Similarly, if a message contains URL, then it could be a spam, so we add some higher weight. In the end, we check if the SMS_weight is greater than a certain spam threshhold, then it is classified as a spam SMS.

Finally, we implement these techniques in an Android application named SMSSpamFilter.

III. Implementation

The Android application SMSSpamFilter detects incoming SMS, reads its contents and determines whether the message is a spam or not. If the SMS is spam, then it is the u silently stored to a different folder instead of Inbox and the mobile user is not notified. Whereas, if it is a legitimate SMS, it is saved to Inbox and the user is notified as usual. The user can then check spam messages in the application if desired.

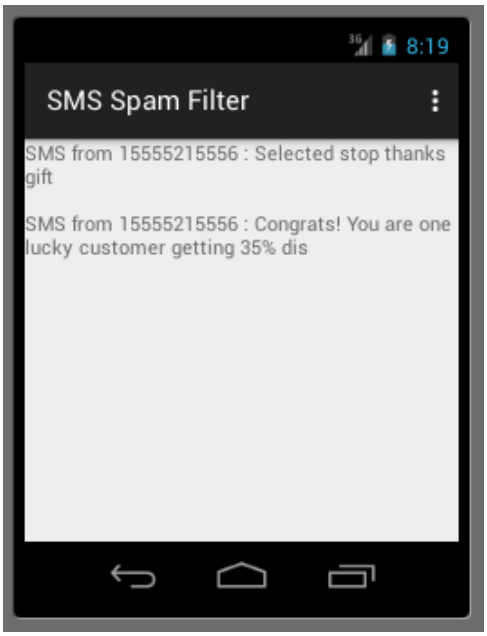

Fig. 1. SMSSpamFilter Android App Screen

To implement this, first of all we provided SMSSpamFilter to use permissions such as RECEIVE_SMS and READ_SMS. Then we defined a BroadcastReceiver which receives broadcast intents and will process if the intent is ”android.provider.Telephony.SMS_RECEIVED” that is when a new SMS is received. To direct this new SMS to the SMSSpamFilter before the default SMS application, we set a higher priority for SMSSpamFilter application. So that if the SMS is a spam it is not forwarded to default SMS application at all.

IV. Experiments and Results

We evaluate the accuracy of the proposed technique using a publicly available SMS collection of 5,574 messages in English, which was introduces in []. These messages were collected by Almeida, T.A., Gómez Hidalgo and available at http://www.dt.fee.unicamp.br/˜tiago/smsspamcollection/. This collection includes 4,827 SMS legitimate messages and 747 spam messages, tagged as spam or ham (legitimate) accordingly.

We created a testing module which classified all these messages as either spam or legitimate and then we compared with the dataset. It was found that the system detects 418 spam messages out of 747 spam messages and identifies 51 messages as spam from 4,827 legitimate messages. Hence, the accuracy of the system is 55.96% and false positive rate is 1.05%. The accuracy can be improved by creating a comprehensive list of spam keywords and false positives can be minimized by training the weights of spam keywords. Also, the spam collection did not include sender numbers for the messages. Thus, the effectiveness of the technique of checking sender number was not actually verified.

V. Discussion

We discuss the cases when the application fails, possible cases of false positives and other more effective spam filtering techniques.

A.

In recent years, some spam messages include URL shortners or URLs without specifying http or https. Such URLs are not detected by our system. Hence SMSs such as ”For the most sparkling shopping breaks from 45 per person; call 0121 2025050 or visit www.shortbreaks.org.uk” could not be detected as spam. This can be handled easily by modifying regular expression for URL or considering different domains and URL shortners.

B.

There are some messages sent from short codes and they are important messages for us for e.g. AT&T bill reminder. But, since the message is sent from short code, our scheme will detect it as a spam SMS. So, this is one case of false positive. There is a possibility that we would miss these important messages. To deal with this problem, we can integrate one more technique which is called white listing. In this method, we would save the number in whitelist and SMSs from the numbers from whitelist will never be detected as spam.

C.

Another disadvantage of the system is that the list of spam keywords is limited as it was created manually. It is static and not comprehensive. We could update it using collection of spam messages and train the weights of the spam keywords accordingly. Another approach is Learning-based content filters or Bayesian filters if we use Bayesian learning methods. In

this approach, filters [2] automatically induce or learn a spam classifier from a set of manually classified examples of spam and legitimate (or ham) messages (the training collection).

VI. Conclusion

In this research project, we studied different spam features such as shorter sender numbers, spam keywords, special characters, URLs and used them to determine whether an SMS is spam or not. We used a list of spam keywords and assigned weight to each of them. We checked whether the sum of all spam features in a message exceeds a defined threshold to classify it as a spam. We implemented this framework in an Android application named SMSSpamFilter. This application filters incoming spam SMS and silently forwards it to a different folder instead of inbox. The users can check these spam messages if desired. The proposed framework was tested on a collection 5,574 text messages in English. It was found that the accuracy of the system was 55.96% and false positive rate was 1.05%. We also discuss when the system fails to detect spam message and how we can improve its accuracy. To avoid important messages received from short codes being detected as spam, we can implement whitelist. The list of spam keywords can be updated and weights can be adjusted for better accuracy by training over larger dataset. Otherwise, we can use Bayesian filters in which filters automatically induce or learn a spam classifier from a set of manually classified examples of spam and legitimate messages.

Acknowledgment

We would like to thank Almeida, T.A., Gómez Hidalgo for making available a collection 5,574 text messages for public use. It helped us to evaluate our system.

References

[1] Gmez Hidalgo, J.M., Cajigas Bringas, G., Puertas Sanz, E., Carrero Garca, F. Content Based SMS Spam Filtering. Proceedings of the 2006 ACM Symposium on Document Engineering (ACM DOCENG'06), Amsterdam, The Netherlands, 10-13, 2006.

[2] Graham, Paul. Better Bayesian Filtering. Proceedings of the 2003 Spam Conference, January 2003.

[3] Alper Kursat Uysal, Serkan Gunal, Semih Ergin, Efnan Sora Gunal. A Novel Framework for SMS Spam Filtering. (IEEE) 2012

[4] Sarah Jane Delany, Mark Buckley, Derek Greene. SMS spam filtering: Methods and data.