2018 15th International ISC (Iranian Society of Cryptology) Conference on Information Security and Cryptology (ISCISC)
August 28–29, 2018; Shahid Rajaee Teacher Training University, Tehran, Iran

SMSBotHunter: A Novel Anomaly Detection Technique to Detect SMS Botnets

Farnood Faghihi
Department of Computer Engineering
Tarbiat Modares University
Tehran, Iran
farnood.faghihi@modares.ac.ir

Mahdi Abadi
Department of Computer Engineering
Tarbiat Modares University
Tehran, Iran
abadi@modares.ac.ir

Asghar Tajoddin
Department of Computer Engineering
Tarbiat Modares University
Tehran, Iran
a.tajoddin@modares.ac.ir

***Abstract*—Over the past few years, botnets have emerged as one of the most serious cybersecurity threats faced by individuals and organizations. After infecting millions of servers and workstations worldwide, botmasters have started to develop botnets for mobile devices. Mobile botnets use different mediums to communicate with their botmasters. Although significant research has been done to detect mobile botnets that use the Internet as their command and control (C&C) channel, little research has investigated SMS botnets per se. In order to fill this gap, in this paper, we first divide SMS botnets based on their characteristics into three families, namely, info stealer, SMS stealer, and SMS spammer. Then, we propose SMSBotHunter, a novel anomaly detection technique that detects SMS botnets using textual and behavioral features and one-class classification. We experimentally evaluate the detection performance of SMS-BotHunter by simulating the behavior of human users and SMS botnets. The experimental results demonstrate that most of the SMS messages sent or received by info stealer and SMS spammer botnets can be detected using textual features exclusively. It is also revealed that behavioral features are crucial for the detection of SMS stealer botnets and will improve the overall detection performance.**

***Index Terms*—anomaly detection; dynamic analysis; one-class classification; SMS botnet**

I. Introduction

The term botnet is used to define a network of compromised machines under the remote control of a human operator called the *botmaster*. The compromised machines are called *bots*, short for robots, reflecting the fact that all bots follow the instructions given by the botmaster [1]. The bots in a botnet communicate with the botmaster through a communication channel called the command and control (C&C) channel.

Botnets are developed to carry out a variety of malicious activities, including but not limited to information theft, launching distributed denial of service attacks, spreading spam, stealing computational resources, monitoring network traffic, and phishing [2], [3]. Financial gain is usually the primary motivation for the design and development of botnets [1].

Early botnets aimed to compromise personal computers solely. Nowadays, however, botnets are also developed for mobile and IoT devices, which have very limited resources (battery, processor, memory, and bandwidth) and encompass some particular features not seen in their counterparts. Therefore, many of the existing botnet detection techniques are rendered inefficient for such devices.

Among the mobile operating systems, Android has topped the market, as it has acquired roughly 85 percent of the worldwide smartphone market share in the first quarter of 2017, according to IDC [4]. As a result, mobile botnets in general and Android botnets in particular as well as their detection techniques have gained a growing interest among research community in recent years [5]–[10].

Although Android was first released in September 2008, the first Android malware, known as FakePlayer.a, was discovered in August 2010. This relatively simple malware pretended to be a movie player with a Microsoft Windows Media Player icon. However, instead of playing movies, it exploited the infected mobile device's texting capabilities to send SMS messages to premium rate numbers without the user's consent or knowledge [11]. From then on, although attackers' motives have not changed much, the strategies used in writing Android malware have continued to evolve toward the employment of sophisticated techniques, such as anti-debugging and code obfuscation [12]. Additionally, the functionality of Android malware has been extended to cover more complex malicious activities other than sending simple SMS messages to premium rate numbers.

At the end of 2010, a new piece of Android malware, known as Geinimi, was discovered. Geinimi is considered to be the first Android malware to display traditional botnet functionalities. It was classified as "the most sophisticated Android malware we have seen to date" by McAfee [13]. Geinimi was repackaged into legitimate apps and consisted of additional classes to add a backdoor-like functionality. It had a list of more than 20 commands implemented in its source code. However, the security companies were not able to see a fully operational C&C server sending commands back to the malware. For this reason, it was not confirmed that the intent

978-1-5386-7582-3/18/$31.00

TABLE I
SOME OF THE COMMANDS SUPPORTED BY TIGERBOT

Command	Description
*[key]*21*	Activate
*#[key]	Deactivate
*[key]*17*a*b	Send SMS to 'a' with content 'b'
*[key]*18	Capture image

of this malware was to build a mobile botnet [13]. Later in 2012, mobile security companies reported the emergence of another piece of Android malware called TigerBot, a rather complicated botnet to date, capable of receiving commands through SMS to perform malicious activities, such as capturing images, sending spam SMS messages, and so on. Table I shows some of the commands supported by this botnet.

In parallel with the outbreak of Android botnets in the wild, the research community showed high interest in further studying the Android botnet phenomenon. For example, Xiang *et al.* [5] introduced an Android botnet, called Andbot, which exploits a C&C mechanism named URL Flux to conceal the botmaster's commands. Later on, Hamandi *et al.* [6] demonstrated how to build and maintain an SMS botnet that targets Android devices specifically. In their design, they defined a simple topology that aims at minimizing the number of SMS messages sent by the botmaster. The topology is a two-level tree where the root is the botmaster and other nodes are bots. The bots at level 1 relay commands by sending SMS messages to the bots at level 2. In addition, Hua and Sakurai [7] proposed a proof-of-concept SMS botnet that abuses SMS messages to propagate the botmaster's commands based on a simple flooding algorithm. They theoretically proved that the uniform random graph is the most efficient topology for this botnet.

In this paper, after inspecting the source code and behavior of different SMS botnets, we group them into three families, namely, info stealer, SMS stealer, and SMS spammer. Then, we define a set of textual and behavioral features, and build two models for SMS messages by training some one-class classifiers on a dataset of benign SMS messages. We evaluate the detection performance of SMSBotHunter using a dataset of SMS messages sent or received by human users as well as SMS botnets.

The rest of this paper is organized as follows: Section II briefly reviews related work. Section III presents SMSBotHunter. Section IV reports experimental results, and finally Section V concludes the paper.

II. RELATED WORK

As Android botnets are becoming more and more sophisticated, the use of machine learning techniques is proving more and more useful in detecting them. Whereas few features might be sufficient for the detection of simple Android malware, a variety of distinctive features should be evaluated for the detection of sophisticated ones.

Feizollah *et al.* [8] proposed three network-level features, namely, connection duration, TCP size, and the number of GET/POST parameters, to learn a classifier for detecting Android malware. Later on, Meng and Spanoudakis [9] proposed a mobile botnet detection system that uses machine learning techniques to detect network traffic generated by mobile botnets.

Although many techniques have been developed for detecting mobile botnets by using their network traffic (as in [8] and [9]), it seems that little research is carried out, focusing on detection techniques for SMS botnets per se.

SMS messages initiated by mobile malware, also known as SIMM messages or malicious SMS messages in this work, are quite similar and, in many cases, they have identical structures, leading to a relatively small space of malicious SMS messages compared to benign ones. Alzahrani and Ghorbani [14] proposed a framework that has the ability to identify SMS botnets in Android devices as they are sent to a central server. The framework provides a model which applies signature detection on smartphone SMS messages and behavioral detection on collected data at the central server. This leads to a response from the decision-and-action module that sends an action to the smartphone to be performed. Later on, Kühnel and Meyer [15] investigated the use of the SVM classifier for classification of malicious and benign SMS messages. To do so, they first collected a set of 155 SMS messages sent by 35 mobile malware samples through static and dynamic analysis, and then they obtained a set of 4,539 benign SMS messages from [16]. By combining these two sets of SMS messages, they created their own dataset. Further, they designed a set of character-based features to build their classification model. Eventually, they applied the SVM classifier with several combinations of designed features to their dataset. The obtained results showed that by using the best feature set, they could achieve a minimum detection rate of 72.06 and a maximum detection rate of 85.54 percent using 5-fold cross-validation. In addition, in their other work [17], they showed that supervised machine learning algorithms can reliably detect malicious SMS messages based on features derived from their content. Further, Johnson and Traore [18] analyzed the inflow of SMS messages using intents and proposed a model to detect non-user initiated and malicious SMS messages on Android devices.

III. OUR WORK AND CONTRIBUTIONS

In this section, we present SMSBotHunter, an anomaly detection technique that uses one-class classification to detect SMS botnets on mobile devices. SMSBotHunter consists of two main steps: training and detection. In the training step, we train two one-class classifiers on a dataset of SMS messages sent or received by human users: one classifier for sent and another for received SMS messages. This results in two models representing human users' behavior while sending and receiving SMS messages. In the detection step, by calculating deviations from the established models, we decide whether or not newly arrived SMS messages are related to an SMS

TABLE II
LIST OF TEXTUAL AND BEHAVIORAL FEATURES

Textual Features	1	Total number of characters
	2	Percentage of whitespace characters
	3	Percentage of slash characters
	4	Percentage of numeric characters
	5	Percentage of lowercase letters
	6	Percentage of uppercase letters
	7	Percentage of special characters
	8	Entropy of characters
	9	Length of the first word
	10	Percentage of numeric characters in the first word
	11	Percentage of uppercase letters in the first word
	12	Percentage of special characters in the first word
	13	Total number of words
	14	Average length of words
	15	Percentage of words with numeric or special characters
	16	Percentage of shorthands
	17	Percentage of meaningless words
	18	Presence of smileys
	19	Presence of URLs
Behavioral Features	20	Input source
	21	Contact type (known or unknown)
	22	Time difference from last SMS message

botnet. The main advantage of SMSBotHunter is that it does not require prior knowledge of SMS botnets and can thus detect new families of them.

The principal challenge in the training step is defining a suitable set of features that are capable to differentiate between benign and malicious SMS messages. By observing the characteristics of malicious SMS messages as well as the behavior of compromised devices when sending or receiving these SMS messages, we define a number of features that can be divided into two main categories: textual and behavioral. While the former characterizes the textual contents of SMS messages, the later profiles user behavior while sending or receiving SMS messages. Our observations show that none of these features alone is enough to discriminate malicious SMS messages from benign ones. Table II shows the list of textual and behavioral features.

A. Textual Features

In order to gain financial profit, many primitive mobile botnets send SMS messages to premium numbers to generate profit. These SMS messages usually contain a number or a short sequence of characters and, thus, differ from those sent by human users. Features 1, 4, 9, 10, and 13 help us detect this type of mobile botnet. Some other mobile botnets, such as TigerBot, are not aimed at making financial gain solely. Such mobile botnets often steal confidential information from the infected device or perform other malicious activities after receiving C&C messages. The C&C commands often contain a single word with numeric and/or special characters that show the command number and, thus, differ from normal English text. Features 4, 7, 9, 10, 12, 13, and 15 helps us distinguish English text from such C&C messages. Additionally, in some mobile botnets, the botmaster and bots encrypt their C&C messages to hide the payload from human users. In such cases, the encrypted messages are more like random texts, which differ significantly from meaningful English texts. The encrypted messages can be detected by measuring their entropy. In English text, some characters are more likely to repeat and, thus, the entropy of English text is usually lower than that of a randomly generated text. Moreover, words are separated by whitespaces and, thus, a certain amount of whitespaces is expected in English text. Additionally, some SMS messages sent by mobile botnets contain meaningless words (such as command numbers in C&C messages and URLs in spam messages). Such words can be detected by checking them against a dictionary. Features 17 and 19 help us distinguish such SMS messages from those sent by human users. On the other hand, some features help us model the behavior of human users. For example, keyboards on mobile devices often capitalize the first character of SMS messages. Feature 11 reflects this behavior. Furthermore, human users do not use special characters or meaningless words very often.

B. Behavioral Features

To improve the accuracy of SMSBotHunter and reduce false alarms, we also consider behavioral features. These features help us distinguish benign and malicious SMS messages by utilizing the user's interaction with the mobile device.

For the sent SMS messages exclusively, we determine their input source, which can be KBD, MSG, or APP. The input source of an SMS message is KBD if it has already been typed using the keyboard, MSG if there is a similar SMS message in the user's inbox, or APP if it is none of the above.

Many malware families such as Zitmo.A and Zitmo.E leak information by stealing SMS messages from the user's inbox and sending it to specific numbers. Such malicious activities are usually done in order to spy sensitive information or defeat two-factor authentication. Inspecting the input source of SMS messages can help us detect such malware.

Other behavioral features can be defined for both sent and received SMS messages. Many mobile botnets reply to specific SMS messages sent by their botmasters. These replies are often automatically generated by the bot app on the user's mobile device. Since generating an automatic reply does not take long, the time gap between receiving an SMS message containing commands and sending its reply is usually very small. This can be used to distinguish malicious replies from legitimate ones. Another distinguishing feature for such replies is that their input source is neither the keyboard nor the user's inbox.

While most of the botnets need to send/receive SMS messages to/from non-contact numbers (their C&C servers) to communicate with the botmaster, human users usually contact others in their contact list. Thus, checking whether or not the sender or recipient of an SMS message is in the contact list

TABLE III
LIST OF SMS BOTNET FAMILIES IN OUR EXPERIMENTS

SMS Botnet Family	Description
Info stealer	Steals personal information based on received commands
SMS stealer	Steals SMS messages from the user's inbox
SMS spammer	Sends spam SMS messages

can be beneficial in distinguishing malicious SMS messages sent by mobile botnets from those sent by human users.

IV. EXPERIMENTS

We implemented a prototype of SMSBotHunter and conducted several experiments to evaluate its performance in detecting SMS botnets. For this purpose, we developed an Android app capable of recording the values of features in Table II upon arrival or departure of SMS messages. We then used this app to observe the behavior of different Android apps, including some SMS botnet apps. On the basis of the insights obtained in this way, we were able to categorize Android SMS botnets into three distinct families, as shown in Table III. Since most of the SMS botnet apps we downloaded could not be run (due to their C&C servers were inactive or blocked), we simulated the behavior of different SMS botnet families as well as human users to create our training and testing datasets.

The text of benign and spam SMS messages was taken from [16], and the list of C&C commands for the info stealer family was taken from [12] as well as those we obtained earlier from our observations. After gathering the texts for benign and malicious SMS messages, we developed a Java application to extract the values of textual features from them. In this application, we considered a word meaningless if it was not an English word, a shorthand, or a smiley.

To simulate the behavioral features of our datasets, we considered statistics of SMS communication patterns. To be more precise, for modeling sent SMS messages, we considered three equiprobable input sources, namely, KBD, MSG, and APP (see Subsection III-B for more information). This was because an SMS message can be typed with the keyboard, copied from another SMS message, or taken from other apps (e.g., a web browser). This feature was absent when building a model for received SMS messages. We also considered the fact that users often send and receive SMS messages to and from phone numbers inside their contact list. Hence, we marked phone numbers in the contact list as known contacts and others as unknown contacts. For simulating the time difference from last SMS message, we considered the communication patterns of 8 users with a single user. The inter-event times between sending consecutive SMS messages were simulated according to [19], [20] using the power-law libraries mentioned in [21].

The simulations allowed us to create a training dataset of 2,000 SMS messages sent or received by human users, and also a testing dataset of 8,700 SMS messages sent or received by human users as well as SMS botnets.

A. Evaluation

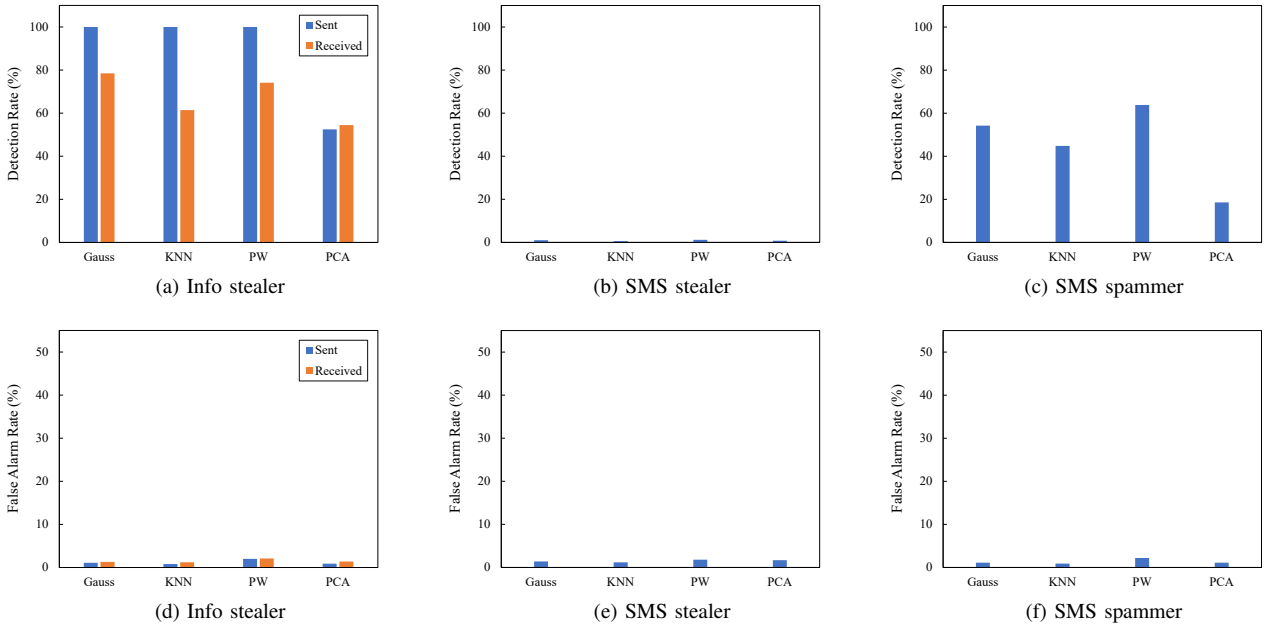
We investigated the usefulness of various features in our feature set. For this purpose, we ranked the features using three common ranking methods, namely, chi-square, gain ratio, and correlation. Tables IV and V report the obtained results for sent and received SMS messages, respectively.

TABLE IV
RANK OF FEATURES FOR SENT SMS MESSAGES

Feature Rank	Chi-Square	Gain Ratio	Correlation
1	8	20	20
2	4	22	21
3	5	21	1
4	1	4	13
5	13	10	8
6	17	8	22
7	20	5	17
8	15	13	2
9	2	1	5
10	6	2	4
11	22	17	15
12	21	14	10
13	14	18	14
14	10	6	6
15	11	15	19
16	7	7	18
17	9	3	9
18	12	19	3
19	16	12	7
20	3	11	12
21	19	16	16
22	18	9	11

TABLE V
RANK OF FEATURES FOR RECEIVED SMS MESSAGES

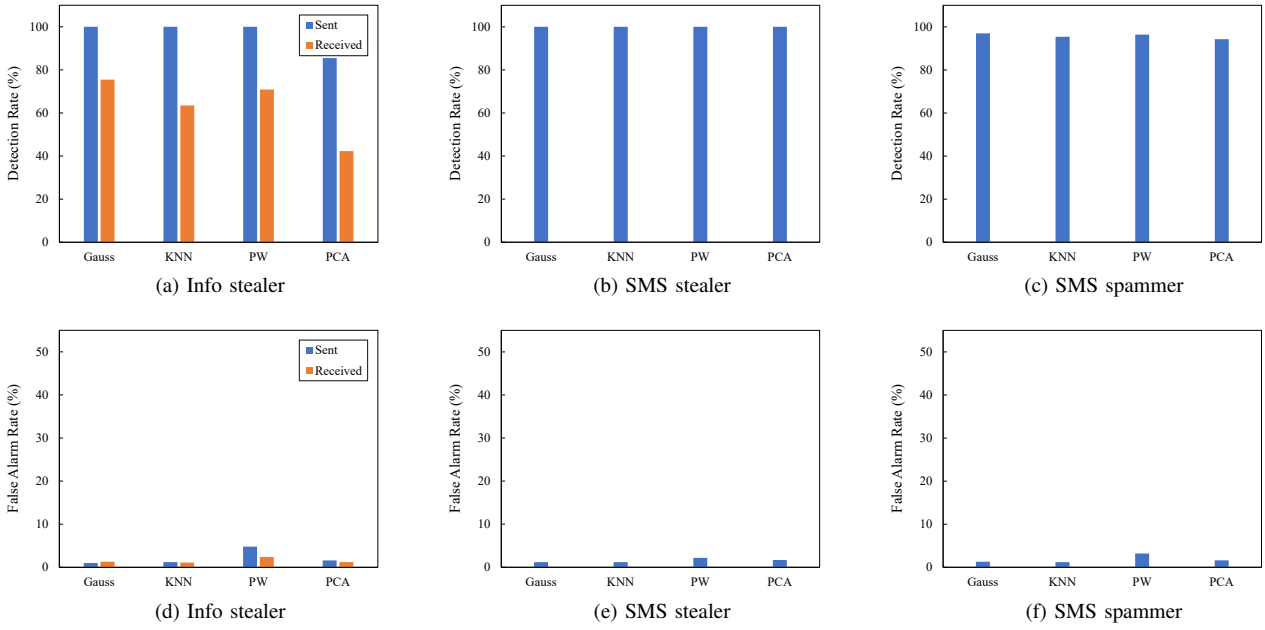
Feature Rank	Chi-Square	Gain Ratio	Correlation
1	8	13	2
2	13	1	8
3	6	6	13
4	1	2	1
5	2	11	17
6	5	8	15
7	11	15	14
8	17	17	4
9	15	5	9
10	7	10	10
11	14	7	5
12	4	4	11
13	12	16	7
14	9	18	12
15	10	14	18
16	16	19	16
17	18	12	19
18	19	9	6
19	21	21	3
20	22	22	21
21	3	3	22



(a) Info stealer (b) SMS stealer (c) SMS spammer

(d) Info stealer (e) SMS stealer (f) SMS spammer

Fig. 1. Detection performance of different models when using textual features only.



(a) Info stealer (b) SMS stealer (c) SMS spammer

(d) Info stealer (e) SMS stealer (f) SMS spammer

Fig. 2. Detection performance of different models when using both textual and behavioral features.

Clearly, from Table IV, we observe that behavioral features (Features 20, 21, and 22) are among the top-ranked features and play a significant role in the detection of malicious sent SMS messages. It is also revealed that some of the textual features (e.g., Features 4 and 8) can improve the detection performance. Further, from Table V, we observe that Features 8 and 13 are among the most discriminative features for detecting malicious received SMS messages. This may be due to the fact that info stealer botnets usually use a limited set of characters in their SMS messages to carry a command to an infected mobile device and, thus, the text of their SMS messages has lower entropy and less number of characters. Also, SMS spammer botnets often send SMS messages that contain a lot of words and URLs to delude users and, thus, their SMS messages have higher entropy and more characters.

In addition, in order to evaluate the detection performance of SMSBotHunter, we built models of human users' behavior while sending and receiving SMS messages using four one-class classifiers, namely, Gauss, KNN, Parzen Window (PW), and PCA. Then, we evaluated these models on our

testing dataset using 10-fold cross-validation. We also tried our experiments with and without behavioral features. We set the targeted false positive rate to 1%. Figs. 1 and 2 show the obtained results in terms of detection rate (DR) and false alarm rate (FPR). From Fig. 1, we observe that all the one-class classifiers could achieve relatively good results on info stealer and SMS spammer botnets using textual features only. This is due to the fact that SMS messages sent or received by these botnets usually differ from those sent by human users. For instance, these SMS messages often contain command codes or spam texts that vary significantly from normal English text (i.e., in length and content). However, all the one-class classifiers show very poor performance in the detection of SMS messages sent or received by SMS stealer botnets while using textual features solely. This is because these SMS messages have the same text as benign SMS messages and, thus, they cannot be distinguished from them using textual features alone. The results in Fig. 2 show that behavioral features could improve the overall detection performance. To be more precise, behavioral features increase the detection rate for SMS stealer botnets notably. The reason for such a significant increase is that SMS messages stolen by this family have similar text to benign ones, but they are different in terms of behavioral features. To elaborate more, it takes a little time for a human user to compose or forward an SMS message, but almost no time for a bot. Thus, by inspecting the input source of an SMS message and the time difference from last SMS message, we can detect automatically generated replies. However, behavioral features could not improve the results on info stealer botnets since received commands have no input source and botmasters sometimes conceal their C&C number in the contact list.

V. Conclusion

In this paper, we presented SMSBotHunter, an anomaly detection technique based on one-class classification for detecting SMS botnets. SMSBotHunter builds models of SMS messages sent or received by human users, and then uses these models to detect SMS messages initiated by SMS botnets.

We categorized SMS botnets into three families, namely, info stealer, SMS stealer, and SMS spammer. Then, we simulated the behavior of each of the SMS botnet families as well as human users to create our training and testing datasets. The training dataset consisted of SMS messages sent or received by human users. Using this dataset, we trained two one-class classifiers, one for sent and another for received SMS messages. We ran our experiments with four one-class classifiers, namely, Gauss, KNN, PW, and PCA, and also different subsets of features. We achieved the best results with the help of behavioral features. These features increased the average detection rate for SMS spammer botnets and played a critical role in detecting SMS stealer botnets. During our experiments, PCA produced poor results and PW produced very good results but with many false alarms. KNN and Gauss usually outperformed other classifiers with Gauss producing the highest detection rate and lowest false alarm rate on average among all the SMS botnet families.

References

[1] R. A. Rodríguez-Gómez, G. Maciá-Fernández, and P. García-Teodoro, "Survey and taxonomy of botnet research through life-cycle," *ACM Comput. Surv.*, vol. 45, no. 4, pp. 45:1–45:33, Aug. 2013.

[2] Z. Zhu, G. Lu, Y. Chen, Z. J. Fu, P. Roberts, and K. Han, "Botnet research survey," in *Proc. 2008 32nd Annual IEEE Int. Computer Software and Applications Conf. (COMPSAC'08)*, Turku, Finland, Jul. 2008, pp. 967–972.

[3] H. Rouhani Zeidanloo, M. Jorjor Zadeh, P. Vahdani Amoli, M. Safari, and M. Zamani, "A taxonomy of botnet detection techniques," in *Proc. 2010 3rd Int. Conf. on Computer Science and Information Technology (ICCSIT'10)*, vol. 2, Chengdu, China, Jul. 2010, pp. 158–162.

[4] "Worldwide smartphone OS market share," http://www.idc.com/promo/smartphone-market-share/os, May 2017.

[5] C. Xiang, F. Binxing, Y. Lihua, L. Xiaoyi, and Z. Tianning, "Andbot: Towards advanced mobile botnets," in *Proc. 4th USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET'11)*, Boston, MA, USA, Mar. 2011, pp. 1–7.

[6] K. Hamandi, I. H. Elhajj, A. Chehab, and A. Kayssi, "Android SMS botnet: A new perspective," in *Proc. 10th ACM Int. Symp. on Mobility Management and Wireless Access (MobiWac'12)*, Paphos, Cyprus, Oct. 2012, pp. 125–130.

[7] J. Hua and K. Sakurai, "Botnet command and control based on Short Message Service and human mobility," *Comput. Netw.*, vol. 57, no. 2, pp. 579–597, Feb. 2013.

[8] A. Feizollah, N. B. Anuar, R. Salleh, F. Amalina, R. R. Ma'arof, and S. Shamshirband, "A study of machine learning classifiers for anomaly-based mobile botnet detection," *Malaysian J. Comput. Sci.*, vol. 26, no. 4, pp. 251–265, Dec. 2013.

[9] X. Meng and G. Spanoudakis, "MBotCS: A mobile botnet detection system based on machine learning," in *Risks and Security of Internet and Systems*, ser. Lecture Notes in Computer Science, C. Lambrinoudakis and A. Gabillon, Eds. Cham, Switzerland: Springer International Publishing, 2016, vol. 9572, pp. 274–291.

[10] G. Kirubavathi and R. Anitha, "Structural analysis and detection of Android botnets using machine learning techniques," *Int. J. Inf. Secur.*, vol. 17, no. 2, pp. 153–167, Apr. 2018.

[11] D. Maslennikov, "First SMS trojan for Android," https://securelist.com/blog/virus-watch/29731/first-sms-trojan-for-android/, Aug. 2010.

[12] R. Nigam, "A timeline of mobile botnets," *Virus Bulletin*, Mar. 2015.

[13] C. A. Castillo, "Android malware past, present, and future," http://www.mcafee.com/us/resources/white-papers/, 2011.

[14] A. J. Alzahrani and A. A. Ghorbani, "SMS mobile botnet detection using a multi-agent system: Research in progress," in *Proc. 1st Int. Workshop on Agents and CyberSecurity (ACySE'14)*, Paris, France, May 2014, pp. 2:1–2:8.

[15] M. Kühnel and U. Meyer, "4GMOP: Mopping malware initiated SMS traffic in mobile networks," in *Information Security*, ser. Lecture Notes in Computer Science, Y. Desmedt, Ed. Cham, Switzerland: Springer International Publishing, 2015, vol. 7807, pp. 113–129.

[16] T. A. Almeida, J. M. G. Hidalgo, and A. Yamakami, "Contributions to the study of SMS spam filtering: New collection and results," in *Proc. 11th ACM Symp. on Document Engineering (DocEng'11)*, Mountain View, CA, USA, Sep. 2011, pp. 259–262.

[17] M. Kühnel and U. Meyer, "Classification of short messages initiated by mobile malware," in *Proc. 2016 11th Int. Conf. on Availability, Reliability and Security (ARES'16)*, Salzburg, Austria, Sep. 2016, pp. 618–627.

[18] E. Johnson and I. Traore, "SMS botnet detection for android devices through intent capture and modeling," in *Proc. 2015 IEEE 34th Symposium on Reliable Distributed Systems Workshop (SRDSW'15)*, Montreal, QC, Canada, Sep. 2015, pp. 36–41.

[19] H. Wei, H. Xiao-Pu, Z. Tao, and W. Bing-Hong, "Heavy-tailed statistics in short-message communication," *Chin. Phys. Lett.*, vol. 26, no. 2, pp. 1–3, 2009.

[20] Z. Zhi-Dan, X. Hu, S. Ming-Sheng, and Z. Tao, "Empirical analysis on the human dynamics of a large-scale short message communication system," *Chin. Phys. Lett.*, vol. 28, no. 6, pp. 1–4, 2011.

[21] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-law distributions in empirical data," *SIAM Rev.*, vol. 51, no. 4, pp. 661–703, 2009.