Hunting Malicious Bots on Twitter: An Unsupervised Approach

Zhouhan Chen(✉), Rima S. Tanash, Richard Stoll, and Devika Subramanian

Rice University, Houston, TX 77005, USA
{zc12,rtanash,stoll,devika}@rice.edu

Abstract. Malicious bots violate Twitter's terms of service – they include bots that post spam content, adware and malware, as well as bots that are designed to sway public opinion. How prevalent are such bots on Twitter? Estimates vary, with Twitter [3] itself stating that less than 5% of its over 300 million active accounts are bots. Using a supervised machine learning approach with a manually curated set of Twitter bots, [12] estimate that between 9% to 15% of active Twitter accounts are bots (both benign and malicious). In this paper, we propose an unsupervised approach to hunt for malicious bot groups on Twitter. Key structural and behavioral markers for such bot groups are the use of URL shortening services, duplicate tweets and content coordination over extended periods of time. While these markers have been identified in prior work [9,15], we devise a new protocol to automatically harvest such bot groups from live Tweet streams. Our experiments with this protocol show that between 4% to 23% (mean 10.5%) of all accounts that use shortened URLs are bots and bot networks that evade detection over a long period of time, with significant heterogeneity in distribution based on the URL shortening service. We compare our detection approach with two state-of-the-art methods for bot detection on Twitter: a supervised learning approach called BotOrNot [10] and an unsupervised technique called DeBot [8]. We show that BotOrNot misclassifies around 40% of the malicious bots identified by our protocol. The overlap between bots detected by our approach and DeBot, which uses synchronicity of tweeting as a primary behavioral marker, is around 7%, indicating that the detection approaches target very different types of bots. Our protocol effectively identifies malicious bots in a language-independent, as well as topic and keyword independent framework in real-time in an entirely unsupervised manner and is a useful supplement to existing bot detection tools.

Keywords: Bot detection · Social network analysis · Data mining

1 Introduction

In recent years, Twitter, with its easy enrollment process and attractive user interface has seen a proliferation of automated accounts or bots [11,13]. While a few of these automated accounts engage in human conversation or provide

© Springer International Publishing AG 2017
G.L. Ciampaglia et al. (Eds.): SocInfo 2017, Part II, LNCS 10540, pp. 501–510, 2017.
DOI: 10.1007/978-3-319-67256-4_40

community benefits [1], many are malicious. We define malicious bots as those that violate Twitter's terms of service [5] including those that post spam content, adware and malware, as well as bots that are part of sponsored campaigns to sway public opinion.

How prevalent are bots and bot networks on Twitter? Estimates vary, with Twitter [3] itself stating that less than 5% of its over 300 million active accounts are bots. Using a supervised machine learning approach with a manually curated set of Twitter bots, [12] estimate that 9% to 15% of active Twitter accounts are bots (both benign and malicious). An open question is how to efficiently obtain a census of Twitter bots and how to reliably estimate the percentage of malicious bots among them. In addition, it is important to estimate the percentage of tweets contributed by these bots so we have an understanding of the impact such accounts have on a legitimate Twitter user's experience. In particular, malicious bots can seriously distort analyses such as [14] based on tweet counts, because these bots cut and paste real content from trending tweets [9].

In this paper, we propose an unsupervised approach to hunt for malicious bot groups on Twitter. Key structural and behavioral markers for such bot groups are the use of shortened URLs, typically to disguise final landing sites, the tweeting of duplicate content, and content coordination over extended periods of time. While the use of shortened URLs and tweeting of duplicate content has been separately identified in prior work [9,15], we devise a new protocol that follows this up by verifying content coordination between bot groups over extended periods of time. Our bot detection protocol has four sequential phases as illustrated in Fig. 1. Our unit of analysis is a cluster of accounts. The initial clustering is based on duplicate text content and the use of shortened URLs. The final detection decision is made by examining the long term behavior of these account clusters and the extent of content coordination between them.

Our experiments with this protocol on actively gathered tweets with shortened URLs from the nine most popular URL shortening services shows a complex picture of the prevalence and distribution of malicious bots. Fewer than 6% of accounts tweeting shortened URLs are malicious bots, except for *ln.is* (27%) and *dlvr.it* (8%). The tweet traffic generated by malicious bot accounts using shortened URLs from *bit.ly*, *ift.tt*, *ow.ly*, *goo.gl* and *tinyurl.com* is under 6%. But malicious bots using *dlvr.it*, *dld.bz*, *viid.me* and *ln.is* account for 13% to 27% of tweets.

The gold standard for confirming bots is suspension by Twitter. However, as noted by [8,10], there is a time lag between detection of bots by researchers and their suspension by Twitter. If we had a reference list of bots, we could give recall and precision measures for our detection approach. In the absence of such a list, we can only provide a precision measure by comparing our bots with those detected by state of the art methods: a supervised learning approach called BotOrNot [10] and an unsupervised technique called DeBot [8]. We show that BotOrNot misclassifies around 40% of the malicious bots identified by our protocol. Unlike BotOrNot, our approach identifies entire bot groups by their collective behavior over a period of time, rather than using decision rules for classifying individual accounts as bots. DeBot is focused on the question of detecting

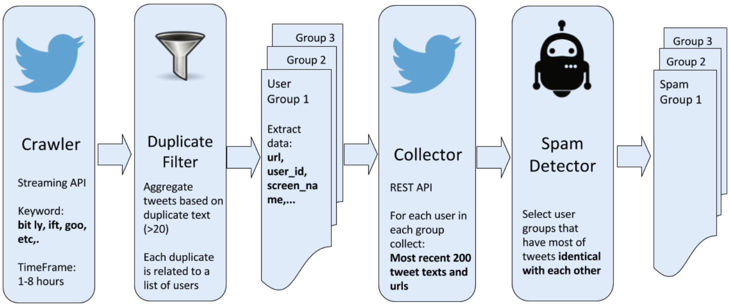
Crawler
Streaming API
Keyword:
bit ly, ift, goo, etc,.
TimeFrame:
1-8 hours
Duplicate Filter
Aggregate tweets based on duplicate text (>20)
Each duplicate is related to a list of users
Group 3
Group 2
User Group 1
Extract data:
url, user_id, screen_name,...
Collector
REST API
For each user in each group collect:
Most recent 200 tweet texts and urls
Spam Detector
Select user groups that have most of tweets identical with each other
Group 3
Group 2
Spam Group 1

Fig. 1. The four phases of our bot detector architecture. Note the second round of tweet collection from a group of accounts that tweet near identical content and which use shortened URLs. This phase analyses the long term tweeting behavior of potential malicious bots.

groups of bots, not necessarily malicious ones, by exploiting coordinated temporal patterns in their tweeting behavior. Since we use duplicated content over a time period, and shortened URLs as primary behavioral markers, in contrast to synchronicity of tweeting, we find a far more diverse group of malicious bots than DeBot. Thus there is only a small overlap (7%) between bot groups found by our protocol and DeBot's, which mainly finds bots associated with news services rather than those that hijack users to spam, ad and malware sites.

In sum, our protocol effectively identifies malicious bots by focusing on shortened URLs and tweeting of near duplicate content over an extended period of time. It is language-independent, as well as topic and keyword independent and is completely unsupervised. We have validated our protocol on tweets from nine URL shortening services and characterized the heterogeneity of the distribution of malicious bots in the Twitterverse.

The remainder of the paper is organized as follows. Section 2 provides brief background on existing Twitter bot detection methods and the novelty and efficacy of our protocol. Section 3 describes and motivates each phase of our bot detection protocol. Section 4 introduces the tweet sets collected from nine URL shortening services used in our experiments, and provides results of our detection protocol on these sets. It also includes results of comparisons of our method with DeBot and BotOrNot. We conclude the paper in Sect. 5 that discusses our main results and directions for future exploration.

2 Background

There is a significant literature on detecting bots on Twitter and other social media forums – recent surveys are in [11,13]. Spam bots on Twitter are constantly evolving, and there is an ongoing arms race between spammers and Twitter's account suspension policies and systems. For instance, link-farming was a dominant category of spam bots in 2012. However, Twitter's new detection algorithms [2] have driven them to extinction.

Current Twitter bot detection methods can be placed in two major categories: ones based on supervised learning which rely on curated training sets of known bots, and unsupervised approaches that need no training data. Supervised methods generally work at the level of individual accounts. They extract hundreds to thousands of features from each account based on properties of the account itself and its interaction with others, and learn decision rules to distinguish bots from human accounts using a variety of machine learning algorithms [7,9,15]. The state of the art in this line of work is BotOrNot [10] which is a Random Forest classifier that uses more than a thousand features to distinguish bots from humans. Features include user profile metadata, sentiment, content, friends, network and timing of tweets. This method has two limitations. First, it makes decisions at the level of individual accounts and therefore fails to identify groups of accounts that act in concert. Second, it works only on accounts tweeting in English, and it is not adaptive; requiring re-training with new human-labeled data as new types of bots emerge.

The state-of-the-art in unsupervised bot detection is DeBot [8], which uses dynamic time warping to identify accounts with synchronized tweeting patterns. This protocol relies on tweet creation time and not its contents. Our results show that DeBot primarily finds news bots with temporally synchronized tweeting patterns. It does not capture malicious spam bots that are temporally uncorrelated, but that tweet duplicate trending content in order to hijack users to spam and malware sites.

Our unsupervised detection method fills a gap in Twitter bot detection research. The underlying features of bots identified by our method are accounts using shortened URLs and near duplicate content over an extended period of time. Because the method is unsupervised, it is not biased by any keyword, topic, hashtag, country or language. It does not require human labeled training sets, and can be deployed in a real time online fashion.

3 Our Methods

Our spam detection method consists of four components run sequentially: crawler, duplicate filter, collector and bot detector. The **crawler** collects live tweets from the Twitter Streaming API using keyword filtering [4]. We choose prefixes of the domain name of a URL shortening service as keywords. The **duplicate filter** selects suspicious groups of accounts for further analysis. It first hashes all tweet content extracted from the *text* field of the tweet's json representation and maps each unique tweet text to a group of users who tweet that content. The filter selects duplicate tweeting groups of size 20 or greater. This threshold of 20 enables us to focus on more significant bot groups. To make sure accounts do in fact violate Twitter's terms of service over a period of time, we perform a second level of tweet collection on each member of a suspicious group. The **collector** gathers the 200 most recent tweets of every account in each suspicious group using Twiter's REST API. This step ensures that we filter out innocent users who happen to tweet a few of the same texts as bots.

The **bot detector** clusters accounts in a group that have most of their historical tweets (200 most recent tweets) identical to each other. Given a group G of n accounts $a_1, \ldots, a_n$, sets $T(a_1), \ldots, T(a_n)$ of tweets where $T(a_i) = \{t_{i1}, \ldots, t_{i200}\}$ of the 200 most recent tweets for each account $a_i, 1 \leq i \leq n$, it constructs the set C of tweets that are tweeted by at least α accounts in the group. That is,

$$t \in C \iff |\{i \mid t \in T(a_i); 1 \leq i \leq n\}| \geq \alpha \tag{1}$$

In the next step, the detector measures the overlap between the tweet set $T(a_i)$ associated with an individual account and the set C of tweets for the group G that account a_i is a member of. The potential bots in the group, denoted by the set S, are identified as follows,

$$a_i \in S \iff \frac{|T(a_i) \cap C|}{|T(a_i)|} \geq \beta \tag{2}$$

Thus there are two parameters in our detection protocol: α, which we call *minimum duplicate factor*, which influences the construction of the most frequent tweet set C, and β, which we call the *overlap ratio*, which determines the ratio of frequent tweets in the tweet set associated with an account. Accounts that meet criteria (1) and (2) for a specific choice of α and β are identified as malicious bots in our protocol. In all of our experiments reported in the next section, we use $\alpha = 3$ and $\beta = 0.6$. These parameters were obtained after cross-validation studies which we do not have space to document here.

4 Experimental Evaluation

4.1 Landscape of Twitter Trending URLs

To justify the use of URL shortening as a marker in our bot detection protocol, we examined the distribution of all URLs in the Twitter stream to estimate the fraction of tweets containing URL shorteners. We first streamed more than thirty million live tweets using keyword *http*, extracted all URLs within each tweet, and sorted them by frequency of occurrence. While this is only a sample of all tweets with embedded URLs, we believe it is an unbiased sample since the Twitter Streaming API does not favor one particular region/language/account over another. Figure 2 shows top trending URLs on Twitter constructed with this sample. Shortened URLs clearly constitute a major fraction of tweet traffic.

4.2 Datasets and Results

Based on our analysis of the distribution of URLs on Twitter, we choose to study nine popular URL shortening services *bit.ly*, *ift.tt*, *ow.ly*, *goo.gl*, *tinyurl.com*, *dlvr.it*, *dld.bz*, *viid.me* and *ln.is*. In our sample, more than 24% of tweets with embedded URLs are generated from these nine service providers.

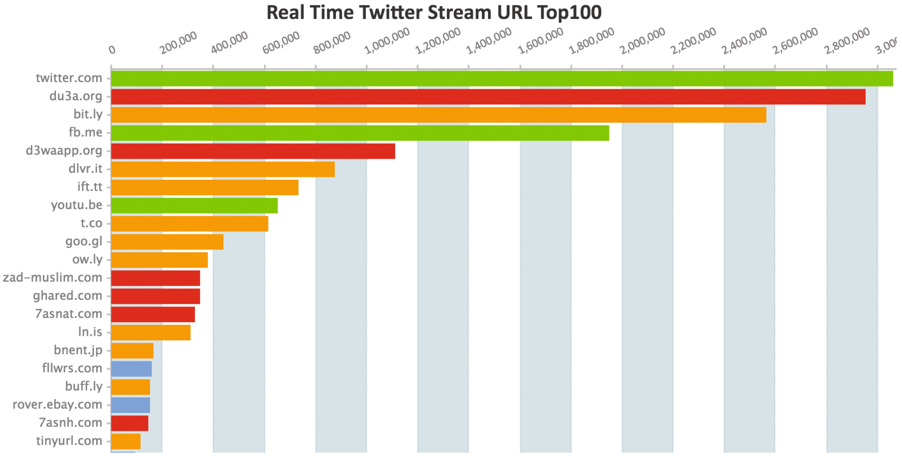

Fig. 2. Top trending URLs on Twitter (Green bars are social media websites, red bars are websites tweeting Quran verses, orange bars are URL shortening services, and blue bars are others. In this paper we focus on the orange bars. For real-time, complete top 100 trending URLs, visit our Twitter Bot Monitor [16].)

Table 1 shows the number and percentage of bot accounts identified by our detection protocol, the percentage of our bot accounts also identified by DeBot, and percentage of accounts that are identified by our protocol and later become suspended by Twitter. We note that the percentages of bot accounts vary greatly among the nine URL shorteners, ranging from 2.4% to 23%, and the percentage of tweets generated by these bot accounts vary from 2.7% to 26.65%, with an average of 10.51%. Four URL shorteners – *dlvr.it*, *dld.bz*, *viid.me* and *ln.is*, have more than 10% of their tweets generated by bot accounts, suggesting that those URL shorteners are more abused by malicious users than the other five.

4.3 Scaling Experiments

In July 2017, we performed a series of experiments on all nine URL shortening services in which we scaled the number of tweets gathered to around 500,000 in order to verify the robustness of our results in Table 1. Two of URL shorteners were no longer available: *ln.is* is suspended [6], while *viid.me* has been blocked by Twitter due to its malicious behavior. Table 2 gives the detailed statistics of our bot detection performance on the remaining URL shortening services.

Compared to our first set of experiments, we see an increase in the percentage of bot accounts in six out of seven URL shortening services, and increased percentages of tweets in all URL shortening services. The percentages of bot accounts range from 5.88% to 17.80%, and the percentage of tweets generated by these bot accounts vary from 14.74% to 56.46%. Together with Table 1, we find that the rate of bot account creation outpaces Twitter suspensions.

Table 1. Statistics of Twitter accounts from nine URL shortening services. Note the uptick in suspended accounts identified by us on `viid.me`

URL shortener	Total # of accts	Total # of bots	% bots we found	% tweets from bots we found	% our bots found by DeBot	% our bots susp. by Twitter until 6/10/17	% our bots susp. by Twitter until 7/17/17
bit.ly	28964	696	2.40%	4.44%	12.93%	3.74%	4.74%
ift.tt	12543	321	2.56%	3.54%	11.21%	2.80%	9.97%
ow.ly	28416	894	3.15%	3.22%	6.04%	45.30%	48.21%
tinyurl.com	20005	705	3.52%	5.70%	1.99%	5.39%	7.66%
dld.bz	6893	304	4.41%	13.36%	10.20%	8.22%	11.84%
viid.me	2605	129	4.95%	21.66%	22.48%	38.76%	55.81%
goo.gl	11250	710	6.31%	2.70%	8.73%	0.42%	3.24%
dlvr.it	15122	1194	7.90%	13.34%	22.86%	7.37%	9.13%
ln.is	25384	5857	23.07%	26.65%	3.57%	1.11%	1.25%

Table 2. Statistics of Twitter accounts from nine URL shortening services (10X scale experiments)

URL shortener service	Total # of accts	Total # of bots	% bots	% tweets from bots
bit.ly	193207	22938	11.87%	16.11%
ift.tt	75024	4415	5.88%	16.70%
ow.ly	182539	31416	17.21%	26.07%
tinyurl.com	49563	4644	9.37%	14.74%
dld.bz	11705	1036	8.85%	56.46%
goo.gl	177030	31515	17.80%	27.88%
dlvr.it	86830	6517	7.51%	18.58%
ln.is	N/A	N/A	N/A	N/A
viid.me	N/A	N/A	N/A	N/A

4.4 Comparison with Existing Bot Detection Methods

Twitter Suspension System. We revisited the account status of bots detected by our protocol to check if they have been suspended by Twitter. The last two columns in Table 1 show percentages of suspended accounts among all bot accounts identified by our protocol, one collected in June and the other in July. As of July, more than 56% of bot accounts using *viid.me* and 48% of accounts using *ow.ly* have been suspended. However, fewer than 15% of bot accounts that use the other seven URL shortening services have been suspended.

DeBot. We compare our results with DeBot in the following manner. For all Twitter accounts in the nine datasets, we queried the DeBot API to determine whether or not the account is archived in its bot database. Table 3 documents the number of bots identified by our method and by DeBot. The intersection of those two groups is small in all cases.

Table 3. Overlap of results from our Bot Detector and DeBot

URL shortener	# bot accts we found	# verified accts we found	# bot accts DeBot found	# verified accts DeBot found	Overlap in accts
bit.ly	605	2	1657	57	91
ift.tt	321	0	989	8	38
ow.ly	894	0	1500	34	55
tinyurl.com	705	0	826	9	14
dld.bz	304	0	473	2	31
viid.me	129	0	515	0	31
goo.gl	710	0	822	9	62
dlvr.it	1194	17	1843	19	281
ln.is	5857	0	2383	7	216

To investigate the low overlap in detected bots, we checked for verified accounts among the ones identified as bots by both methods. To determine the percentage of news bots (defined as an account that tweets at least once with a URL from a list of established news media URLs), we used the Twitter REST API to collect the 200 most recent tweets from these accounts. Table 4 shows that more than 50% of bot accounts identified by DeBot are news bots, compared to 15% based on our method.

Table 4. Comparison of percentage of news accounts

Protocol	Total bots	News bots	% News bots
Our protocol	696	102	14.66%
DeBot	1748	947	54.18%

Thus, what DeBot finds, but our method does not are news bots linked to large news media accounts. What both methods find are bot groups that tweet highly synchronously with duplicate content, and what our method finds but DeBot does not are bot groups using shortened URLs that do not tweet simultaneously.

BotOrNot. We also compared our results with BotOrNot, a supervised, account based Twitter bot classifier. BotOrNot assigns a score of 0 to 1 to an account based on more than 1000 features, including temporal, sentimental and social network information [10]. A score close to 0 suggests a human account,

Table 5. Bot accounts scores from BotOrNot

URL shorteners	Average Score	% bots with score in [0.4,0.6]
bit.ly	0.50	50.65%
ift.tt	0.56	50.46%
ow.ly	0.52	40.83%
tinyurl.com	0.56	66.08%
dld.bz	0.71	17.75%
viid.me	0.68	24.81%
goo.gl	0.53	68.99%
dlvr.it	0.49	44.43%
ln.is	0.44	56.81%

while a score close to 1 suggests a bot account. Like DeBot, BotOrNot also provides a public API to interact with its service. Table 5 shows the statistics of BotOrNot scores of all bots we identified in the nine datasets. In 5 out of the 9 datasets, more than 50% of the scores of accounts identified as bots by our protocol fall in the range of 0.4 and 0.6. We expect scores of bots detected by our protocol to exceed 0.6, so we interpret these results as misclassifications by BotOrNot.

5 Conclusions

In this paper we present a Twitter bot detection method that hunts for a specific class of malicious bots using shortened URLs and tweeting near duplicate content over an extended period of time. Our unsupervised method does not require labeled training data, and is not biased toward any language, topic or keyword. Arguably our method does not capture the most sophisticated bots out on Twitter, yet it is surprising that between 4% to 23% of the accounts we sampled from the streaming API satisfy our bot criteria and remain active on Twitter, generating 4% to 27% of tweet traffic. In the absence of identified bot lists, we resort to comparisons with two of the best bot detection protocols on Twitter to evaluate the effectiveness of our approach. Our work gives us a more nuanced understanding of the demographics of Twitter bots and the severity of bot proliferation on Twitter. Bot removal is a necessary step in any analysis relying on raw counts of collected tweets, so our work is useful for anyone with a Twitter dataset from which duplicates of trending tweets generated by malicious bots need to be eliminated. Our future work involves devising better approaches to evaluating bot detection tools, and developing new criteria for discovering more sophisticated bots that contaminate the Twitter stream. Malicious bot detection is an arms race between bot makers and social media platforms and we hope our work contributes to the design of better bot account detection policies.

References

1. Earthquakebot. https://twitter.com/earthquakebot?lang=en. Accessed 30 Mar 2017
2. Fighting spam with botmaker. https://blog.twitter.com/2014/fighting-spam-with-botmaker. Accessed 20 Mar 2017
3. Twitter Annual Report. http://files.shareholder.com/downloads/AMDA-2F526X/4335316487x0xS1564590-17-2584/1418091/filing.pdf. Accessed 22 Apr 2017
4. Twitter developer documentation. https://dev.twitter.com/streaming/overview/request-parameters. Accessed 20 Mar 2017
5. The Twitter Rules. https://support.twitter.com/articles/18311. Accessed 13 Jan 2017
6. We need you to help to get linkis back to work. http://blog.linkis.com/2017/06/02/we-need-you-to-help-to-get-linkis-back-to-work. Accessed 19 July 2017
7. Cao, C., Caverlee, J.: Detecting spam URLs in social media via behavioral analysis. In: Hanbury, A., Kazai, G., Rauber, A., Fuhr, N. (eds.) ECIR 2015. LNCS, vol. 9022, pp. 703–714. Springer, Cham (2015). doi:10.1007/978-3-319-16354-3_77
8. Chavoshi, N., Hamooni, H., Mueen, A.: DeBot: Twitter bot detection via warped correlation. In: Proceedings of the 16th IEEE International Conference on Data Mining (2016)
9. Chu, Z., Gianvecchio, S., Wang, H., Jajodia, S.: Detecting automation of Twitter accounts: are you a human, bot, or cyborg? IEEE Trans. Dependable Secure Comput. **9**(6), 811–824 (2012)
10. Davis, C.A., Ferrara, V.E., Flammini, A., Menczer, F.: BotOrNot: a system to evaluate social bots. In: Companion to Proceedings of the 25th International Conference on the World Wide Web, pp. 273–274. International World Wide Web Conferences Steering Committee (2016)
11. Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The rise of social bots. Commun. ACM **59**(7), 96–104 (2016)
12. Ferrara, O.V.E., Davis, C.A., Menczer, F., Flammini, A.: Online human-bot interactions: detection, estimation, and characterization. arXiv preprint (2017). arXiv:1703.03107
13. Jiang, M., Cui, P., Faloutsos, C.: Suspicious behavior detection: current trends and future directions. IEEE Intell. Syst. **31**(1), 31–39 (2016)
14. Montesinos, L., Rodrguez, S.J.P., Orchard, M., Eyheramendy, S.: Sentiment analysis and prediction of events in Twitter. In: 2015 CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), pp. 903–910, October 2015
15. Wang, D., Navathe, S., Liu, L., Irani, D., Tamersoy, A., Pu, C.: Click traffic analysis of short URL spam on Twitter. In: 2013 9th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), pp. 250–259. IEEE (2013)
16. Twitter Bot Monitor project on github. https://github.com/Joe--Chen/TwitterBotProject