Proceedings of the Thirteenth International AAAI Conference on Web and Social Media (ICWSM 2019)

Detecting Social Bots on Facebook in an Information Veracity Context

Giovanni C. Santia, Munif Ishad Mujib, Jake Ryland Williams
Department of Information Science,
College of Computing and Informatics,
Drexel University,
30 North 33rd Street, Philadelphia, PA 19104
{gs495,mim52,jw3477}@drexel.edu

Abstract

Misleading information is nothing new, yet its impacts seem only to grow. We investigate this phenomenon in the context of social bots. *Social bots* are software agents that mimic humans. They are intended to interact with humans while supporting specific agendas. This work explores the effect of social bots on the spread of misinformation on Facebook during the Fall of 2016 and prototypes a tool for their detection. Using a dataset of about two million user comments discussing the posts of public pages for nine verified news outlets, we first annotate a large dataset for social bots. We then develop and evaluate commercially implementable bot detection software for public pages with an overall F_1 score of 0.71. Applying this software, we found only a small percentage (0.06%) of the commenting user population to be social bots. However, their activity was extremely disproportionate, producing comments at a rate more than fifty times higher (3.5%). Finally, we observe that one might commonly encounter social bot comments at a rate of about one in ten on mainstream outlet and reliable content news posts. In light of these findings and to support page owners and their communities we release prototype code and software to help moderate social bots on Facebook.

Introduction

The potency of online misinformation campaigns has made itself well known in recent times. Weaponizing this spread of misleading information, an unknown number of human-mimicking automatons have been designed to infiltrate various popular social media venues, generating content in an algorithmic fashion (Ferrara et al. 2016). This automated content is often designed to support a particular agenda or ideology. Bots which operate in this fashion are often referred to as *social bots*. While talk of social bots and misleading information have dominated recent headlines, the problem is not as new as many may think. There is evidence that during the 2010 U.S. Midterm Elections, social bots were used by various parties in an attempt to sway the results (Ratkiewicz et al. 2011). The role bots played in the 2016 U.S. Presidential Election and the Brexit vote (Howard, Kollanyi, and Woolley 2016) has dredged this discussion up to the surface. The unprecedented reach and power that social media networks have has created an ideal ecosystem for the spread and evolution of social bots (Ferrara et al. 2016). Moreover, with an increasing number of people obtaining their news from these sites (Gottfried and Shearer 2016), politicians and their associated campaigns are bolstering their utilization of social media as a means to engage potential voters.

Social bots pose a growing threat, yet the exact nature and size of the problem is not well-known. Estimates for the percentage of automated accounts vary, but a commonly accepted figure puts between 9% and 15% of Twitter users as bots (Varol et al. 2017b). This is an enormous population, exerting considerable gravitas on the ebb and flow of online discussion, making the possibility for impact outside of the online domain ever larger.

The largest factor enabling the emergence and evolution of social bots is the difficulty in detecting them (Ferrara et al. 2016). With social bot designers taking every opportunity to make their creations look as "human" as possible, the detection of social bots poses serious challenges for filtration. At the same time, bot detection is now an essential piece of the social media experience for the average user (Yang et al. 2019). The bot deceit includes the replication of text generated by actual humans, in addition to more sophisticated and clandestine methods capable of interacting with legitimate users. So, an ongoing battle between bot-detection researchers and bot creators has led to a digital arms race. Steps towards an increased capability for bot detection would have far-reaching consequences.

A concerning issue for the social bot detection community is the lack of gold-standard data. While data and work have been developed for Twitter, no such data sets have been designed to aid the detection of social bots on Facebook. The work we describe is an effort towards addressing this issue. With a developed social bot data set, we will seek to produce a detection algorithm capable of elucidating their presence on Facebook.

Further differentiating social bot studies on Twitter and Facebook is the fact that bots are not explicitly forbidden on Twitter, while they are on Facebook. The mere existence of social bots on the Facebook platform is both deceptive and troubling. This issue reared its head in the form of the Cambridge Analytica data collection scandal. It is no secret that the trust levels of the public in the Facebook platform took a massive hit in the aftermath of this scandal. Making progress

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

in the elimination of social bots would go a long way in rebuilding the trust of Facebook's users in the platform, thus there would be much interest from the public in performing the type of work we set out to do.

During the completion of this work, Facebook altered the functionality of their Graph API—without any warning—and made it much more difficult to obtain user-identifying data from public posts, perhaps in response to their public relations meltdown. Our pre-2018 data access allowed us to construct data and prototype social bot detection software for commercial application on Facebook. However, data for implementation now will require page owners to either run our software under their own applications or provide access to our (eventual) implementation of a server-to-server Facebook application with Public Page Content authorization. Thus, we provide an in-page evaluation and set-up for page-owner use, along with a cross-page evaluation to simulate the performance of a potential Facebook app.

To fill the Facebook data gap, this work builds off of a data set—*BuzzFace*—created using a piece of investigative journalism performed by BuzzFeed, in which a selection of news headlines posted to Facebook during September 2016 were annotated for credibility (Santia and Williams 2018). The Facebook community contributed discourse around the posted headlines, resulting in *BuzzFace* containing over 1.6 million comments made by over $800,000$ users. From these users we approach our task through a stratified sample of $1,000$, making a training set tractable for annotation. The completion of this work led to our empirical findings that up to 0.6% of the frequent commentators on reliable news stories in our dataset were social bots with nearly one in ten of the comments on said news stories being created by these bots.

Existing Work

The widespread nature of the social bot problem coupled with the focus of the media on their existence and use has led to their detection being an active domain of research. An informative survey is provided in (Ferrara et al. 2016). A taxonomy of social bot detection systems is proposed which separates the potential methods into three categories:

- systems relying on network theory,
- systems employing crowd-sourced human analysis, and
- machine-learning models using user-based features.

Graph and Crowd-Sourcing Algorithms

Some notable examples of bot detection methods employing graph techniques include (Chowdhury et al. 2017; Wang 2010; Boshmaf, Beznosov, and Ripeanu 2013; Abokhodair, Yoo, and McDonald 2015). These results are nearly uniformly restricted to Twitter bot detection. Facebook has long used a technique in-house involving their social graph to attempt to protect users from social bots, called *Facebook Immune System* (Stein, Chen, and Mangla 2011). An obvious flaw with techniques such as these lies in their over-reliance on the idea that humans interact mostly with other humans. Truly devious agents can take advantage of the social graph to position their bots far away from other bots and avoid detection.

One of the most strictly protected types of data on Facebook is that relating to the users themselves—particularly the data necessary to create social graphs of friends and interactions. Now, after the Cambridge Analytica scandal, this data is sure to only get even more difficult to obtain. So, even if Facebook is able to leverage these data to support their community it is by and large not available to the research community. Thus, we move forward with developing a technique that does not rely on social graphs, and can be applied equally well across platforms.

While crowd-sourcing techniques pose additional promise for detection, they come with the weakness of being potentially less accurate than techniques using experts. A study using these techniques is detailed in (Wang et al. 2012). Novices and experts were employed to look at Facebook profiles and assess whether they represented real people or not. The process their experts used was very close to the process which our annotators used. There are a few additional examples of this technique being used, one of the more interesting ones being (Ratkiewicz et al. 2011).

Feature-Based Machine Learning

Several studies applying user-based features to machine learning methods on Twitter have been undertaken. An influential bot detection framework—called *SentiBot*—is described in (Dickerson, Kagan, and Subrahmanian 2014). A diverse ensemble of user features existing in four categories were used: tweet syntax, tweet semantics, user behavior, and network-centric properties. The model was constructed using the *India Election data set*, a large corpus of Tweets related to the 2014 Indian Election. One study presents extremely impressive results (Ji et al. 2016). This work first delves into detail about the possible techniques bots might employ in order to evade detection and then establishes 9 new features that hope to detect these evasive techniques. These features were appended to 9 features that were already in use for bot detection. Ultimately, this method yielded an F_1 score of 96.3% when applied to Twitter. While the study goes into copious detail concerning the evasion methods and features used in classification, the descriptions of the data sets used and the algorithms employed are less clear. Thus, little information was gleaned that may have helped improve the results in our work. The most influential study in formulating our algorithm was (Clark et al. 2016). Here a similar feature-based approach was taken in identifying social bots on Twitter. Other notable examples of studies which applied feature-based machine learning algorithms to Twitter social bot detection are discussed further in (Paavola et al. 2016; Stringhini, Kruegel, and Vigna 2010; Chu et al. 2010).

Given the timely and important nature of the rise of social bots, the problem lends itself well to competitions. DARPA held a challenge called *The DARPA Twitter Bot Challenge* in early 2015 to detect bots which promoted specific topics that they referred to as "influence bots" (Subrahmanian et al. 2016). The three top-placing teams all used feature-based machine learning algorithms to produce their results, with their feature categories including: tweet syntax, tweet

semantics, temporal behavior features, user profile features, and network features. These techniques were overall very similar to those of (Dickerson, Kagan, and Subrahmanian 2014), albeit the final results had lower scores.

The first publicly accessible Twitter social bot detection system was called *Bot or Not?* and is discussed in (Davis et al. 2016; Varol et al. 2017a). To use the service, the user must first input a Twitter username that they wish to investigate. The system then uses the Twitter REST API to obtain the necessary metadata and data. The authors state that over $1,000$ features are collected by the system in the following categories: network, user, friends, temporal, content, and sentiment. This system boasts excellent and comprehensive results, but is unfortunately limited only to detecting bots on Twitter. Presumably the largest contributing factor to the dominance of Twitter in the bot detection domain is plethora of data it provides to researchers. The vast majority of the $1,000$ features employed by the *Bot or Not?* system could not be used in our own work on Facebook, as they were simply unavailable to us. For example, all of the social graph features are impossible to replicate using Facebook data, as obtaining a list of a user's friends is impossible using the Facebook Graph API (unless you happen to be friends with said user). Despite these differences, this system is a good point of comparison to our work as we similarly intend for our algorithm to provide continual updates in a streaming fashion.

Dataset

The corpus used throughout this study consists of $1,684,093$ Facebook comments obtained using the Facebook Graph API, included in the *BuzzFace* data set (Santia and Williams 2018). These comments were created by a total of $843,690$ unique Facebook users. They comprise the entirety of the Facebook community's discussion of $2,282$ specific news articles posted over a period of seven consecutive business days in September 2016—the height of the U.S. Presidential Election. These news articles were the focus of a piece of investigative journalism by BuzzFeed (Silverman et al. 2016), where a team of journalists analyzed each and assigned them a veracity categorization from among the following categories: *mostly true*, *mostly false*, *mixture of true and false*, and *no factual content*. In order to keep the analysis balanced, BuzzFeed tracked the articles from nine different news outlets of differing political leanings: mainstream, left-leaning, and right-leaning. The mainstream outlets were *ABC News Politics*, *CNN Politics*, and *Politico*, while the left-leaning outlets were *Addicting Info*, *Occupy Democrats*, and *The Other 98%*, and finally the right-leaning outlets consisted of *Eagle Rising*, *Freedom Daily*, and *Right Wing News*. It is important to note that all of the chosen outlets have been "verified" by Facebook and thus maintain quite an influential and respected position among the various news entities on the social media platform.

In order to create and fine-tune our model, it was necessary to first construct an annotated subset of the data. Thus, we produced manual annotations on comments made by $1,000$ of the users. The volume of comments made per user varies quite dramatically throughout the set, but the majority of users are only responsible for a single comment. Thus, establishing an annotation set required stratified sampling based on user comment frequency. We first partitioned the $843,690$ users into bins of size ten based on comment volume. Since sampling evenly-spaced bins would over-represent the minimal commenters and under-represent the most frequent commenters, we finally selected 100 log-spaced bins to establish the set of users.

	Agreement	**Cohen's Kappa**
Total	86%	55.46%
High buckets	74%	48.76%
Low buckets	97%	38.52%
Binary total	88.4%	62.54%

Table 1: Annotation metrics for various subsets of the dataset. The high buckets had most of the non-human users, and was often difficult to decide between cyborg or spammer, leading to the lower Kappa value than the set at large. The low buckets were overwhelmingly human users, but because of this fact any disagreement between the annotators drastically impacted the Kappa value. This led to lower value, which fits in with the difficulty in annotating these users as there was much less text to work with. Combining the spammer and cyborg classes into a single bot class significantly improved inter-annotator agreement, and this classification is what was used by the algorithm in the end.

Annotation

Following a bot annotation protocol based on Twitter (Clark et al. 2016), project team members attempted to assign each user to one of the following three categories: *human*, *spammer*, or *cyborg*. *Spammers* were defined as those primarily automating recurrent posts, often with slight variations to elude account suspension. *Cyborgs* were defined as accounts which primarily copy/paste pieces of text and strategically direct them together in an attempt to pass as human. The annotation protocol also included a *robot* category, whose members generated text independently in response to external stimuli, such as environmental readings, e.g., weather bots. This work excludes the robot category from its annotation since Facebook prohibits these types of obvious automatons. Incidentally, none were observed in the dataset over the course of this work.

Two expert annotators each went through the dataset's stratified sample. After this, a list of users for which the two disagreed was formulated. The two annotators then came together and went through each of the users on the disagreement list and performed a second round of annotations to come up with a final set of agreed upon annotations for modeling. In the end, the annotations created by the separate annotators disagreed on 14.0%. More detailed analysis of the inter-annotator agreement is provided in Table 1.

Upon completion of their task, the annotators determined 84.7% of the users sampled to be human and 15.3% some kind of social bot. The platform exhibited a small spammer population (7 in the $1,000$), especially in comparison to its

cyborgs (146 in the $1,000$). Since differences in volumes between spammers and cyborgs were insufficient to allow for separate detection, this work's modeling proceeds with a simple binary distinction between humans and social bots (of either type). Under this binary distinction, annotator disagreement dropped from 14.0% to 11.6%, resulting in an increased Cohen's Kappa value of 62.54, i.e., while there was some confusion over assignment of bot type, there was less confusion over an account's status as *some* kind of automation. Ultimately, the development of a tool that distinguishes spammers from cyborgs would likely require annotation of a much larger sample.

In the context of the political discussions in this work's data, the key differences observed between the spammers and cyborgs was that of motive and practicality on the Facebook platform. Spammers were few in number. Repeatedly posting a phrase or slogan without being entirely obvious to Facebook's bot prohibition is challenging. Yet, one spammer was found who solely posted the following comment: "Wow." This all contrasted with the cyborgs, who appeared to analyze the content of an article or headline and regurgitate a vague but semi-relevant message, often enticing redirection to an external link. While these behavioral observations are consistent with those made in the protocol's inception on Twitter (Clark et al. 2016) we view the reduced presence of spammers on Facebook to be a result of the platform's prohibition.

Overall, the process of annotation was challenging. For the most active users, there was an ample amount of text for the annotators to analyze. Here the text was first searched for direct repetition of comments—the most obvious type of automated content. Another clue was often the recurrent posting of similar links. An additional helpful indicator was the time-stamp associated with each message. Frequently accounts were observed to have created content—"typed"—faster than a human conceivably could.

With fewer comments, classification based on the dataset alone was more difficult for the less active users. However, the dataset (in its originally-accessed form) entailed a user ID of each commenter, and thus in these cases the annotators were able to look up the Facebook profile of the users and analyze their publicly-facing information for signs of automation. The annotators assigned "human" labels to accounts with more complete profiles. For example, profiles with several pictures (of the same person), a regular posting pattern, or having several active friends and family members were consistently labeled human. Alternatively, profiles with only one or two photos and no evidence of close friends or family were reviewed with more suspicion. There is much literature on the topic of how to identify bots and/or fake profiles on social media. Two highly-detailed and informative pieces on the topic which support our methodologies are (Shaffer 2018; Australia 2017). Alas, as Varol et al. put it in the highly influential and seminal (Varol et al. 2017a): "there is no simple set of rules to assess whether an account is human or bot". A difficulty which arises when using data acquired by looking at Facebook profiles is the presence of private profiles. These profiles provide almost no information to users which are not friends with the user in question. The only data which could be acquired from these profiles included the profile picture, name of the user, and location. The annotators chose to still follow the above-mentioned classification flow, which meant that these users were nearly always determined to be bots due to the lack of information regarding posting pattern, family members, etc. This may have caused a bias in the annotation process had there been a prevalence of private profiles, but only a tiny number of private profiles were encountered. Sometimes the collection of comments made by a user were partially suspicious, and also seeming to be those of a real person, but the user's Facebook profile appeared normal. Other times there were users that showed signs of being a social bot textually, but they too had very normal-looking profiles. In these difficult cases the annotators leaned towards the evidence on the profiles.

While the overall 15.3% social bot population found seems to fit in well with many major assessments of the proportions of automated users versus human users in other social media platforms, such as Twitter (Varol et al. 2017b), it is in fact an overestimate of Facebook's overall social bot presence. Social bots generally post often, and our stratified sample of $1,000$ users was intentionally directed to a hyper-posting population. This ensured a dataset having social bot activity sufficiently rich for development. For an analysis of the (much lower) total presence of social bots on Facebook in this work's data, see the Evaluation section and Fig. 2 in the results of the trained model's application.

EDA and Feature Development

With its close guard on platform data, viability for bot detection software on Facebook depends closely on its ability to work with minimal data. Even when posts and comments are public, the identities of users and their connections are generally not available. Under the current Graph API version (3.2), a page owner or authorized server-to-server application will generally be restricted to obtaining post and comment content, timestamps, and user identities (only in certain circumstances). Thus, this work explores features derived largely from text and timestamps. Exploratory data analysis and consideration of existing research directed our approach to the following measures:

- *average response time* ($\bar{t}$): The commentary in Facebook threads follows a very particular structure: users may leave comments on the actual post itself (comments which we have dubbed as *top-level*), or they may leave replies to these top-level comments (which we refer to as *replies*). For any top-level comment except the first in a thread, the response time measures how long it has been since the posting of the previous top-level comment in that thread. In the case that the top-level comment is the first in the thread, we treated the initial news post itself as the previous comment. For any reply aside from the initial reply to a top-level comment, the response time measures how long it has been since the previous reply was made. In the case that a reply is the first on a top-level comment, we treat the parent top-level comment as the previous reply. These are all measured in number of seconds.
- *average comment length* ($\overline{C}$): The annotators noticed that

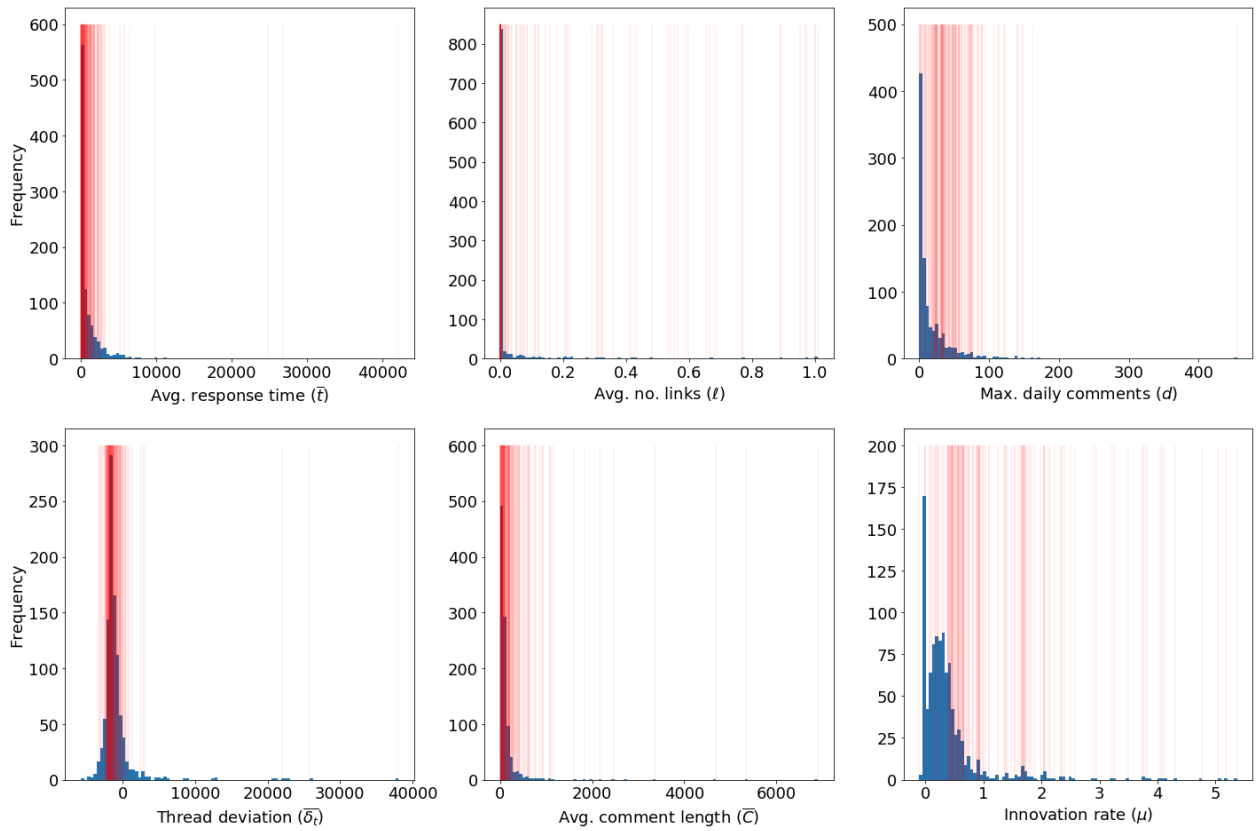

Figure 1: Histograms of the six parameters used in the developed model. Blue distributions represent all users, while red vertical lines represent the values of social bots in these distributions.

users deemed social bots tended to make comments that were long and dense. We decided this may be a useful feature for classification, measuring length by the number of total characters. The only ambiguity here is what to do when the comment in question is just an emoticon. When using the Facebook Graph API to obtain such a comment, the object that returns has an empty string as its message key value. Thus all comments made with only an emoticon were assigned length 0. Perhaps if future versions of the Graph API allow for retrieval of these comments, more detailed analysis may be performed with them.

- *innovation rate* (μ): Known for its regularity as a characterization of human language production, this measure has previously been used with success to detect social bots (Clark et al. 2016) and is based on word occurrence statistics. μ parameterizes a model describing the rate at which a human utilizes novel words to convey ideas. The innovation rate is modeled with a negative power law, sometimes referred to as the decay-rate exponent. When a user's content is drawn from multiple sources μ characterizes just how "mixed" the content is (Williams et al. 2015), with more mixed content leading to higher values of μ.
- *maximum daily comments* (d): This parameter measures the tendency of an individual to post many comments at once in a cluster. We imagined human users will be unlikely to exhibit large commenting spurts. Pathologically, a bot could be used once or twice to post a massive number of comments, but then rarely ever post again. This activity would not be represented by a simple average of comments made per day. To address this lack of sensitivity, user comments were instead grouped into calendar days to calculate the maximum daily comments.
- *number of links* (ℓ): Regular expressions were used to search the text of each comment for standard HTTP links. Once these were collected, the average number of links per comment was calculated. For humans these values were quite often zero. Despite being low overall, these numbers were observed to be higher for bots, making ℓ potentially useful as a feature.
- *thread deviation* ($\overline{\delta_t}$): We hypothesized that the ebb and flow of the rhythm of human conversation in a single thread might be chaotic, while for bots follow a uniform distribution. To capture this, for each comment, we looked up the thread that it was made in and calculated the average response time of all the comments in the thread. Then we took the difference between the response time of the comment in question and this average—which we called

the thread deviation.

Fig. 1 shows histograms for each of these features. The average response time $\bar{t}$ behaved as hypothesized. The mean response time among the humans was 1568.70 seconds while it was 1086.50 among the social bots. It is quite evident from Fig. 1 (top left) that the social bots were heavily clustered near 0. Thus, for our model we took a higher average response time as support for a user being a human. The average number of links ℓ also behaved as we had postulated; the mean number of links posted per comment by humans was 0.0181, while the same figure for the social bots was 0.1467. As can be seen in Fig. 1 (top middle), the social bots are fairly spread out between 0 and 1 unlike the clustering for the average response time. We took frequent posting of links as an indicator for social bots. The maximum daily comments d was another feature that acted as expected. The mean maximum daily comments among the human users was 11.56, while it was 47.85 for the social bots. Looking at Fig. 1 (top right) we see a somewhat different pattern to that of the rate of link posting, with social bots more clustered. We decided thus a high maximum daily comment rate to be an indicator for social bots. The average deviation from the thread mean response time $\overline{\delta_t}$ had some surprising results. The average value of this feature for the human users was -1501.68 seconds while for social bots it was -693.94 seconds. Thus humans responded much more quickly than the average user on a given thread. Observing Fig. 1 (bottom left), we do see some dense clustering of the bots near the middle of the distribution, making outliers representative of human behavior. The mean comment lengths (Fig. 1 (bottom middle)) $\overline{C}$ appeared to fit our hypotheses well. The overall average for human-generated comments was 89.96, while among social bots it was an incredible 408.75. Thus, longer comment length was taken as an indicator for being a social bot. Finally, the values of μ also behaved as expected. The average value of μ for the humans was 0.2544, while it was 1.284 for the social bots. Looking at Fig. 1 (bottom right) we observe that the bots were fairly spread out, with the vast majority of them occurring after $\mu = 0.4$. Thus, we resolved that any μ values within this clustering might be an indicator for a user being a social bot.

Model

This work is an early step into bot detection on Facebook. More complex modeling would be possible—especially if richer data were more accessible from the platform—but we explore the limits of a simple model based on capture regions. Each feature is a single numeric value for each user. However, we experiment with features in two distinct roles. As can be observed in Fig. 1, each of μ, d, ℓ, and $\overline{C}$ (see the EDA section for descriptions) appear to separate bots (red vertical lines) away from the central range of the full feature distributions. Thus, we leverage them as bot *indicators*. Through similar observation we note that $\bar{t}$ and its variant, $\overline{\delta_t}$, appear to cluster bots tightly, with the overall distributions having higher variation. So, while this second group of features would be inadequate for indicating bots with reasonable precision, bot-anomalous values could be leveraged to prevent the model from making some false positives, if implemented as *circuit breakers*. Features referred to as *indicators* are strictly used to identify bots, while the *circuit breaker* features are used to identify a user as a human with the highest priority. No matter how many indicator features show us that a user should be classified as a bot, if a single circuit breaker feature tells us that the user is human, then the user will be classified as a human.

For each indicator I, we define a capture region r_I, of positive (bot) prediction as the compliment of an interval centered on the training set's human-average feature value $\overline{I}$, for some tuned window size, Δ_I. In other words, a user is predicted as positive if its indicator value I, falls outside of $r_I = [\overline{I} - \Delta_I, \overline{I} + \Delta_I]$ for any I.

This indication method is equivalent to that used in (Clark et al. 2016), which produced favorable classification performance. However, we seek to use the identified circuit breakers to boost precision and further improve the model. So, as a natural extension of the capture-region framework, we define an exclusion region of negative (human) prediction, for each circuit breaker B. An exclusion region r_B, is centered on the average feature value for the bots in the training set $\overline{B}$, for some tuned window size Δ_B. So, if $I \notin r_I$ for any I, a positive prediction is imminent, and the circuit breakers are checked, whereupon if $B \notin r_B$ for any B, the positive prediction is ignored and left as negative.

Experimental Design

The assessment of a user's status hinges upon the measurement of our four indicators and two circuit breakers. However, many of these features are highly dependent on the number of comments made by the user by the time of assessment. In a streaming classification context, we expect more and more content to become available. So, for all experiments we tune five separate values of each parameter—one for each quintile of users, according to the distribution of number of comments made. In addition to the expectation for model agility, this aspect also allows for the model to be re-applied to a user's data to update its evaluation as data accrues. Thus, all together, our most complex model relies upon four capture and two exclusion regions for each of five groups, i.e., a total of 30 features.

We designed a 10-fold cross validation training scheme with the objective of learning optimal window sizes for each feature, in models defined by various combinations of indicators and circuit breakers. Training was conducted on a randomly-selected 50% of the annotated users, with the other 50% set aside as a blind test set. Window sizes for each indicator were optimized with all combinations of one, two, and no circuit breakers. Four final experiments (one for each circuit-breaker combination) were conducted with *all* indicators present.

Since full parameter scans are not feasible (due to massive computational cost) we explored feature spaces via random sampling. In particular, for each of 10 permutations of a parameter-optimization order (we perform optimization on each parameter one at a time, and we refer to the order in which we perform this as parameter-optimization order),

Tf-idf baselines

	P	R	F_1
NB-M	**70.63** (**57.63**)	**49.66** (**32.96**)	**58.32** (**41.94**)
NB-B	46.22 (37.1)	42.47 (43.5)	44.27 (40.05)
SVM	65.71 (54.32)	25.4 (8.65)	36.64 (14.92)

Extracted feature baselines

	P	R	F_1
RF-G	74.08 (**72.1**)	50.44 (**50.13**)	60.02 (**59.14**)
SVM	35.13 (48.01)	46.01 (40.65)	39.84 (44.03)
RF-E	**76.83** (75.64)	**50.83** (48.27)	**61.18** (58.94)
DTree	55.63 (61.82)	56.98 (56.17)	56.3 (58.86)

Table 2: Classification performance for both extracted feature and tf-idf based baselines for both user- and page-level (parentheticals) scenarios.

we checked 100 randomly selected feature-specific window-sizes from $1,000$ evenly-spaced grid points in the range of a given parameter's values. The collection of window-size parameters that produced the best evaluation metric score on a fold's 10% held-out users determined its best tuned model. The mean of the best parameters for each fold was used to produce a finalized model, which was applied to the held out 50% of users set aside as a blind test set. The evaluation results from the finalized model on the blind test set are those reported in this work. Finally, each model was cross-validated twice with parameters optimized according to the F_1 and $F_{0.5}$ metrics, where the latter was used to produce precision boosted models.

As originally conceived in (Clark et al. 2016), this work's model was applied to the entire collection of comments made by a user. However, following our original data collection Facebook modified their API and access to their data. So, while we were originally able to access the target data—public page content for arbitrary pages with user comments and identities—these data will now only be accessible with page owner authorization. Thus, if page owners apply our tools they will only have access to user comments and identities on their pages. For this real-world implementation case we refine the project's $1,000$ annotated users to a collection of just over $1,500$ user-page posting histories. A user-page posting history refers to a record of a specific user posting a specific page. Thus, a user posting on two different pages would result in two separate user-page posting histories. We refer to the original model evaluation and tuning as the *user-level experiment* (on $1,000$) and this second one as the *page-level experiment*. While the user-level experiment produces a model that would require authorization from many page owners to track user behaviors across pages, the page-level experiment produces a model that can be easily applied to data from only a single page, i.e., with a single page owner's authorization, making the page-level work crucial for any early commercial implementation.

Baselines

Several conventional machine learning systems were utilized for two types of baseline systems—those using text-based features and those leveraging our extracted numeric features. The goal of developing these systems was to obtain a good estimation of the capabilities of available tools for the Facebook bot detection task. We used the Scikit-learn library (Pedregosa et al. 2011) for each of these standard algorithms.

For text-based classification, we used the standard Term Frequency-Inverse Document Frequency (tf-idf) weighting for word frequencies. These tf-idf features were computed by combining all comments posted by a given user (or given user in a page in page-level experiments) into a single document.

Tf-idf classification experiments were run using 3 algorithms: Bernoulli Naïve Bayes (NB-B), Multinomial Naïve Bayes (NB-M), and Linear Support Vector Machine (SVM). For the NB classifiers, we set $\alpha = 0.01$. For the SVM, we used the ℓ_1 norm penalty and a 10^{-3} stopping criteria tolerance for feature selection, and the ℓ_2 norm penalty and a 10^{-4} tolerance for classification.

We took the mean of the results from 5 different random 10-fold cross validation experiments, splitting the annotated users (or user-page combinations) into training (50%) and blind test (50%) sets. Among the 3 tf-idf classifiers, Multinomial Naïve Bayes performed best in both the user- and page-level analysis, with average F_1 scores of 58.32 and 41.94, respectively. The performance of tf-idf based classifiers is listed in Tab. 2.

For the extracted features, we trained entropy and Gini coefficient optimized random forest classifiers (E-RF and G-RF), a decision tree (DTree) classifier, and a linear SVM classifier. For the SVM, we used the ℓ_2 norm penalty and a 10^{-4} tolerance for stopping criteria. The results are listed in Table 2, with the random forest classifiers demonstrating the best performance numbers at both the user (RF-G: $F_1 = 59.14$) and page levels (RF-E: $F_1 = 61.18$).

Evaluation and Discussion

Our model's results in application to the the dataset's blind test set users (50%) are recorded in Tab. 3 in descending order of F_1 score. Ultimately, by this measure the user-level model incorporating all indicators, but only one breaker, $\overline{\delta_t}$, proved best. The resulting performance of this model: $(P, R, F_1) \approx (80.6, 64.29, 71.52)$ functioned quite well, beating the best baseline (random forest) by more than 10 F_1 points.

Reviewing Tab. 3 we note several important observations about the relative impacts of the features. The application of circuit breakers appeared to affect the expected impact of increased precision only in the user-level experiments. Excluding $\overline{\delta_t}$ resulted in an F_1 drop of 4.5 points from the best model, with precision notably 15% lower. However, both of the best page-level models utilized no circuit breakers, with the best precision-enhanced page-level model utilizing only μ. In general, the page-level models appeared to not be helped by the circuit breakers, and made better use of $\bar{t}$ than

Optimization over F_1

I	B	P	R	F_1
$*$	-	**75.4** (65.17)	**63.76** (69.05)	**69.09** (67.05)
μ	-	70.87 (75.36)	60.4 (61.9)	65.22 (67.97)
$*$	$\overline{t}$	71.54 (72.97)	59.06 (64.29)	64.71 (68.35)
μ	$\overline{t}$	70.49 (76.56)	57.72 (58.33)	63.47 (66.22)
$*$	$\overline{\delta_t}$	76.29 (**80.6**)	49.66 (**64.29**)	60.16 (**71.52**)
μ	$\overline{\delta_t}$	75.0 (80.36)	48.32 (53.57)	58.78 (64.29)
$*$	$*$	75.0 (80.0)	48.32 (61.9)	58.78 (69.8)
μ	$*$	73.47 (73.53)	48.32 (59.52)	58.3 (65.79)
$\overline{C}$	$\overline{t}$	54.04 (60.82)	58.39 (70.24)	56.13 (65.19)
$\overline{C}$	-	50.83 (58.49)	61.74 (73.81)	55.76 (65.26)
$\overline{C}$	$\overline{\delta_t}$	50.97 (63.1)	53.02 (63.1)	51.97 (63.1)
$\overline{C}$	$*$	51.33 (63.64)	51.68 (66.67)	51.51 (65.12)
ℓ	-	46.95 (60.58)	51.68 (75.0)	49.2 (67.02)
ℓ	$\overline{t}$	47.97 (61.29)	47.65 (67.86)	47.81 (64.41)
ℓ	$\overline{\delta_t}$	48.2 (60.47)	44.97 (61.9)	46.53 (61.18)
ℓ	$*$	48.12 (58.89)	42.95 (63.1)	45.39 (60.92)
d	-	41.95 (58.06)	48.99 (64.29)	45.2 (61.02)
d	$\overline{\delta_t}$	45.83 (62.82)	44.3 (58.33)	45.05 (60.49)
d	$\overline{t}$	42.11 (57.95)	48.32 (60.71)	45.0 (59.3)
d	$*$	42.41 (64.0)	44.97 (57.14)	43.65 (60.38)

Optimization over $F_{0.5}$

I	B	P	R	F_1
$*$	-	80.0 (69.57)	53.69 (57.14)	64.26 (62.75)
μ	-	**81.25** (78.57)	**52.35** (52.38)	**63.67** (62.86)
$*$	$\overline{t}$	77.78 (78.57)	51.68 (52.38)	62.1 (62.86)
μ	$\overline{t}$	81.32 (81.48)	49.66 (52.38)	61.67 (63.77)
μ	$*$	84.93 (83.33)	41.61 (47.62)	55.86 (60.61)
$*$	$*$	79.75 (77.08)	42.28 (44.05)	55.26 (56.06)
μ	$\overline{\delta_t}$	83.33 (**89.36**)	40.27 (**50.0**)	54.3 (**64.12**)
$*$	$\overline{\delta_t}$	78.57 (81.4)	36.91 (41.67)	50.23 (55.12)
$\overline{C}$	-	79.03 (73.53)	32.89 (29.76)	46.45 (42.37)
$\overline{C}$	$*$	69.44 (74.36)	33.56 (34.52)	45.25 (47.15)
d	-	44.12 (60.61)	40.27 (23.81)	42.11 (34.19)
$\overline{C}$	$\overline{t}$	76.79 (72.73)	28.86 (28.57)	41.95 (41.03)
d	$\overline{t}$	46.55 (65.71)	36.24 (27.38)	40.75 (38.66)
d	$*$	45.87 (64.52)	33.56 (23.81)	38.76 (34.78)
d	$\overline{\delta_t}$	45.05 (66.67)	33.56 (21.43)	38.46 (32.43)
$\overline{C}$	$\overline{\delta_t}$	70.91 (71.88)	26.17 (27.38)	38.24 (39.66)
ℓ	-	62.5 (60.78)	13.42 (73.81)	22.1 (66.67)
ℓ	$\overline{\delta_t}$	65.52 (60.67)	12.75 (64.29)	21.35 (62.43)
ℓ	$*$	72.0 (61.63)	12.08 (63.1)	20.69 (62.35)
ℓ	$\overline{t}$	62.07 (62.5)	12.08 (65.48)	20.22 (63.95)

Table 3: Cross-validated performance for the mean-centered classifier optimized over F_1 and $F_{0.5}$ (boosting precision) for several configurations of indicators (**I**) and circuit breakers (**B**). Here, "$*$" refers to all and "-" refers to none. Precision (**P**), recall (**R**), and $\mathbf{F}_1$ scores are reported as percentages from the blind, held-out test set. Parenthetical numbers refer to the user-level experiments, while the main percentages refer to the page-level experiments and have F_1 values sorting the tables from high to low. Best models according to optimization metrics are highlighted in bold.

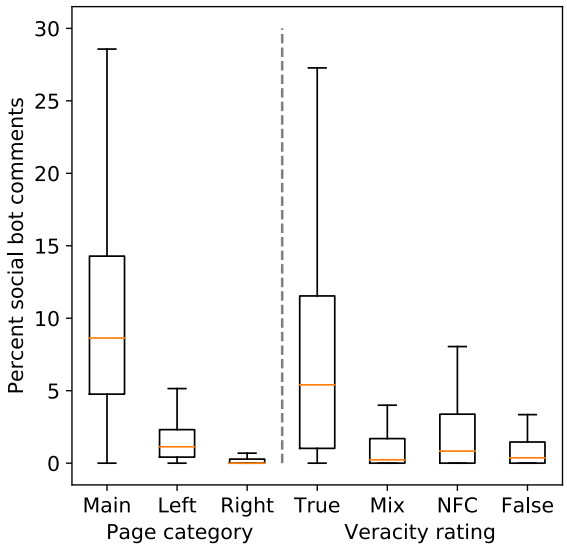

Figure 2: Box plots depicting the percentage of annotated users labeled as social bots present in the commentary threads by (left) media outlet political orientation category and (right) BuzzFeed's veracity annotation category. Outliers are removed so as to allow focus on the comparison of the distributions.

$\overline{\delta_t}$, raising the possibility that $\overline{\delta_t}$'s precision-enhancing value is only accessible with comprehensive data on user posting patterns. Comparing the best F_1-optimized user-level model (all indicators using $\overline{\delta_t}$) to its counterpart using $\overline{t}$, we see an increase of about 8 precision points for using $\overline{\delta_t}$ over $\overline{t}$.

A logical next step in examining our results was the application of our model to the complete dataset. Doing so, our algorithm's application yielded a "bot rate" of 0.06% overall. While this may seem minuscule compared to the most commonly accepted estimates for the same such statistic on Twitter—detailed in (Varol et al. 2017b)—of between 9% and 15% of users being social bots, there is a key difference between Twitter and Facebook. Twitter does not explicitly disallow bots on their platform, while Facebook goes to great lengths to ensure profiles represent actual people. Since Twitter's bot policy is far more lax, the average user can expect to encounter a significant number of automated profiles. Facebook specifically forbids bots to operate on the network, and thus the mere existence of them is an act of deception. Additionally, as stated above, more and more people are getting their daily news from Facebook, and Facebook is arguably more visible to the average user than Twitter is, which makes even the smallest social bot population on Facebook a troubling discovery.

To explore the existence of bot commenting trends we examined the data through the lens of BuzzFeed's post veracity ratings and news outlet political categorization. Trends can be observed by viewing box plots of the threads in Fig. 2, which are quantified by the ratios of comments made by detected social bots in each. The majority of the bot content appears to be focused on *mostly true* rated posts emerg-

ing from pages categorized as mainstream. These results show that threads near the median exhibited bot-produced content at a rate of 10% in these page/post contexts. It is particularly alarming to find elevated percentages on posts with the *mostly true* rating made by mainstream-categorized pages, as these contexts received the most attention in the dataset (Santia and Williams 2018).

Observing these commenting trends in Fig. 2, we also note some contrast with the findings of other social bot research. Shao et al. found in (Shao et al. 2017) that social bots on Twitter seem to mostly focus on sharing content from outlets with very low credibility. On the other hand, Vosoughi et al. (Vosoughi, Roy, and Aral 2018) also examined the spread of false news on Twitter and determined that false and true news are spread by bots at the same rate. They argue that false news is actually spread more often by humans than by bots. However, our results do not analyze the agendas of the social bots (or humans) being detected. So, work analyzing sentiment and social support levels will potentially constitute important future directions. Clearly, there is much work to be done in this domain, and understanding how bots contribute to the spread of information remains an important avenue for future research.

A significant result of this work is the data annotation and subsequent observation of social bot prevalence across the news outlets and information veracity categories. These ground-truth observations hint at the potential targeted nature of social bot application towards reliable content on mainstream news outlets during the height of the 2016 U.S. Presidential Election. Thus, the content being targeted by social bots may be less focused on misinformation, at least directly. Instead, the larger portion of social bot activity in the dataset was directed towards more truthful posts, opening the possibility for their primary strategy being oriented towards undermining truth. Investigating the validity of this hypothesis must be the subject of further investigation, but these results warrant such work. The large-scale application of our classifier to Facebook's comment threads both going back in time and now, streaming, into the future will be an important step forward towards understanding how information is being manipulated online.

Limitations

During the completion of this work Facebook modified a number of their data access and app development policies. Originally and as conceived, anyone with a Facebook account would have been able to utilize our software to monitor social bots on any public Facebook page for their own benefit. Under the present policies, third party applications must 1) request the right to collect public page data from the platform and 2) obtain authorization from each account owner who wishes to monitor the bots on their page. So while any current utilization of our software will require direct implementation by Facebook page owners (page-level models), our software's passage through the platform's app review process will allow our implementation of (user-level) models in a server-to-server application requiring only authorization by the page owners who wish to utilize our tools. While these use cases currently exist, the one originally conceived—allowing any Facebook user to clarify the social bots on any public page—will not be possible under the current conditions. Perhaps as the platform's in-house mitigation efforts continue and their data access policies continue to change, it will be possible for our work on social bots to be extended to user-facing tools. It is also important to note that the data used to train and test the model was strictly from a small set of Facebook news pages, and as such it may take more work and data to generalize the model to Facebook conversations at large.

References

Abokhodair, N.; Yoo, D.; and McDonald, D. W. 2015. Dissecting a social botnet: Growth, content and influence in twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*, 839–851. ACM.

Australia, C. 2017. How to spot a fake facebook account. *McAfee*.

Boshmaf, Y.; Beznosov, K.; and Ripeanu, M. 2013. Graph-based sybil detection in social and information systems. In *Advances in Social Networks Analysis and Mining (ASONAM), 2013 IEEE/ACM International Conference on*, 466–473. IEEE.

Chowdhury, S.; Khanzadeh, M.; Akula, R.; Zhang, F.; Zhang, S.; Medal, H.; Marufuzzaman, M.; and Bian, L. 2017. Botnet detection using graph-based feature clustering. *Journal of Big Data* 4(1):14.

Chu, Z.; Gianvecchio, S.; Wang, H.; and Jajodia, S. 2010. Who is tweeting on twitter: human, bot, or cyborg? In *Proceedings of the 26th annual computer security applications conference*, 21–30. ACM.

Clark, E. M.; Williams, J. R.; Jones, C. A.; Galbraith, R. A.; Danforth, C. M.; and Dodds, P. S. 2016. Sifting robotic from organic text: a natural language approach for detecting automation on twitter. *Journal of Computational Science* 16:1–7.

Davis, C. A.; Varol, O.; Ferrara, E.; Flammini, A.; and Menczer, F. 2016. Botornot: A system to evaluate social bots. In *Proceedings of the 25th International Conference Companion on World Wide Web*, 273–274. International World Wide Web Conferences Steering Committee.

Dickerson, J. P.; Kagan, V.; and Subrahmanian, V. 2014. Using sentiment to detect bots on twitter: Are humans more opinionated than bots? In *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference on*, 620–627. IEEE.

Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; and Flammini, A. 2016. The rise of social bots. *Communications of the ACM* 59(7):96–104.

Gottfried, J., and Shearer, E. 2016. News use across social media platforms 2016.

Howard, P. N.; Kollanyi, B.; and Woolley, S. 2016. Bots and automation over twitter during the us election. *Computational Propaganda Project: Working Paper Series*.

Ji, Y.; He, Y.; Jiang, X.; Cao, J.; and Li, Q. 2016. Combating the evasion mechanisms of social bots. *computers & security* 58:230–249.

Paavola, J.; Helo, T.; Sartonen, H. J. M.; and Huhtinen, A.-M. 2016. The automated detection of trolling bots and cyborgs and the analysis of their impact in the social media. In *ECCWS2016-Proceedings fo the 15th European Conference on Cyber Warfare and Security*, 237. Academic Conferences and publishing limited.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12:2825–2830.

Ratkiewicz, J.; Conover, M.; Meiss, M. R.; Gonçalves, B.; Flammini, A.; and Menczer, F. 2011. Detecting and tracking political abuse in social media. *ICWSM* 11:297–304.

Santia, G., and Williams, J. 2018. Buzzface: A news veracity dataset with facebook user commentary and egos. *International AAAI Conference on Web and Social Media*.

Shaffer, K. 2018. Spot a bot: Identifying automation and disinformation on social media. *Medium*.

Shao, C.; Ciampaglia, G. L.; Varol, O.; Flammini, A.; and Menczer, F. 2017. The spread of fake news by social bots. *CoRR* abs/1707.07592.

Silverman, C.; Strapagiel, L.; Shaban, H.; Hall, E.; and Singer-Vine, J. 2016. Hyperpartisan facebook pages are publishing false and misleading information at an alarming rate.

Stein, T.; Chen, E.; and Mangla, K. 2011. Facebook immune system. In *Proceedings of the 4th Workshop on Social Network Systems*, 8. ACM.

Stringhini, G.; Kruegel, C.; and Vigna, G. 2010. Detecting spammers on social networks. In *Proceedings of the 26th annual computer security applications conference*, 1–9. ACM.

Subrahmanian, V.; Azaria, A.; Durst, S.; Kagan, V.; Galstyan, A.; Lerman, K.; Zhu, L.; Ferrara, E.; Flammini, A.; and Menczer, F. 2016. The DARPA twitter bot challenge. *Computer* 49(6):38–46.

Varol, O.; Ferrara, E.; Davis, C.; Menczer, F.; and Flammini, A. 2017a. Online human-bot interactions: Detection, estimation, and characterization. *International AAAI Conference on Web and Social Media*.

Varol, O.; Ferrara, E.; Davis, C. A.; Menczer, F.; and Flammini, A. 2017b. Online human-bot interactions: Detection, estimation, and characterization. *arXiv preprint arXiv:1703.03107*.

Vosoughi, S.; Roy, D.; and Aral, S. 2018. The spread of true and false news online. *Science* 359(6380):1146–1151.

Wang, G.; Mohanlal, M.; Wilson, C.; Wang, X.; Metzger, M.; Zheng, H.; and Zhao, B. Y. 2012. Social turing tests: Crowdsourcing sybil detection. *arXiv preprint arXiv:1205.3856*.

Wang, A. H. 2010. Don't follow me: Spam detection in twitter. In *Security and Cryptography (SECRYPT), Proceedings of the 2010 International Conference on*, 1–10. IEEE.

Williams, J. R.; Bagrow, J. P.; Danforth, C. M.; and Dodds, P. S. 2015. Text mixing shapes the anatomy of rank-frequency distributions. *Physical Review E* 91:052811.

Yang, K.-C.; Varol, O.; Davis, C. A.; Ferrara, E.; Flammini, A.; and Menczer, F. 2019. Arming the public with ai to counter social bots. *arXiv preprint arXiv:1901.00912*.