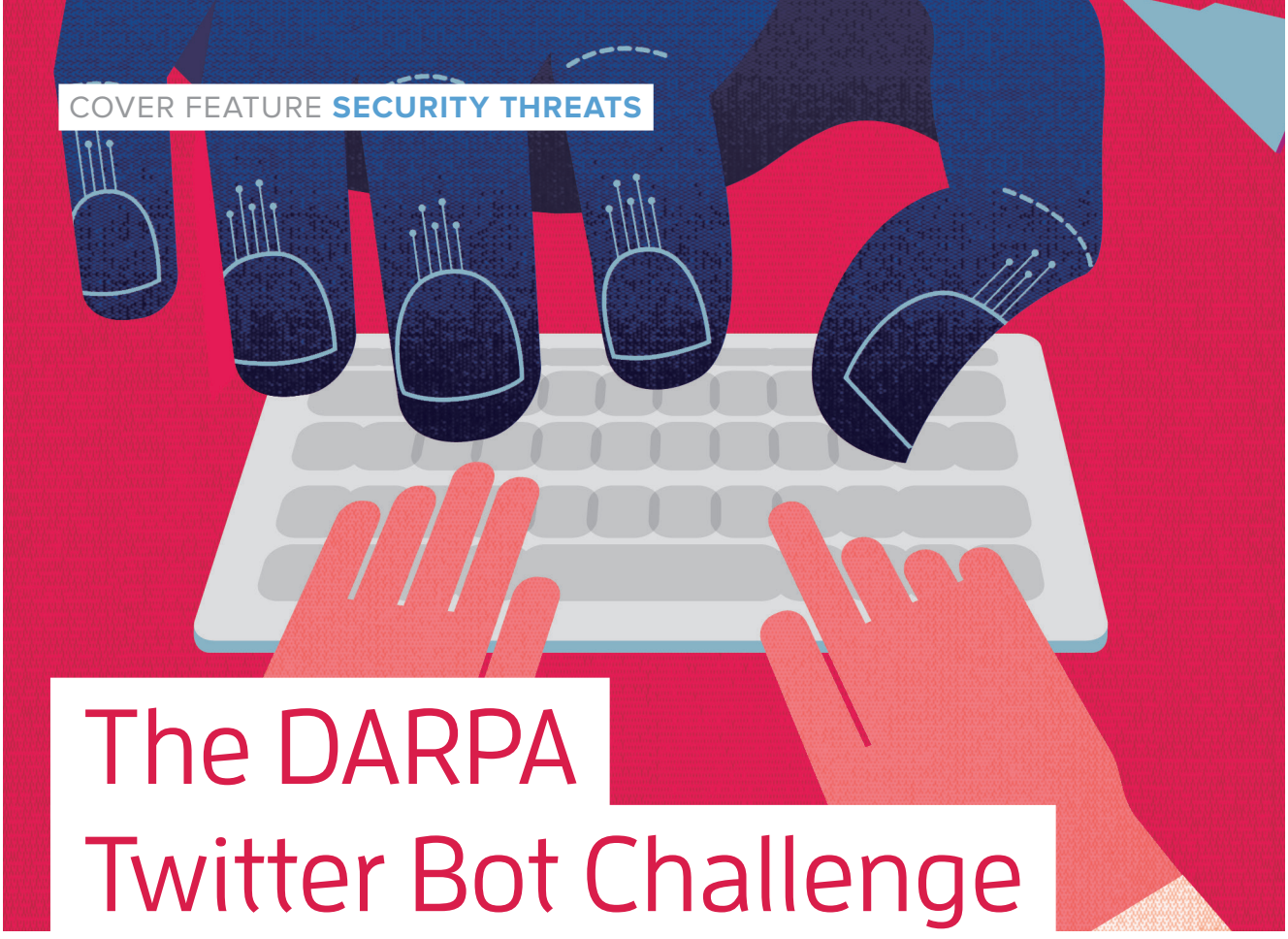

The DARPA Twitter Bot Challenge

V.S. Subrahmanian, University of Maryland
Amos Azaria, Carnegie Mellon University
Skylar Durst and Vadim Kagan, SentiMetrix
Aram Galstyan, Kristina Lerman, and Linhong Zhu, University of Southern California
Emilio Ferrara, Alessandro Flammini, and Filippo Menczer, Indiana University

From politicians and nation states to terrorist groups, numerous organizations reportedly conduct explicit campaigns to influence opinions on social media, posing a risk to freedom of expression. Thus, there is a need to identify and eliminate "influence bots"—realistic, automated identities that illicitly shape discussions on sites like Twitter and Facebook—before they get too influential.

According to a recent US Securities and Exchange Commission filing by Twitter, approximately 8.5 percent of all Twitter users are bots. Many of these bots serve a commercial purpose (such as spambots, which spread spam on various topics, or paybots, which copy content from respected sources and paste it in micro URLs that pay the bot creator for directing traffic to that site), but some are influence bots—realistic, automated identities that illicitly shape discussions on social media sites like Twitter and Facebook, posing a risk to freedom of expression. For example, the terrorist group ISIS used social media to spread radicalism by influencing youth to embrace their cause;[1] an opinion piece in *Forbes* asserted that Russia waged a social media disinformation campaign in the aftermath of Russian actions in the Ukraine;[2] and computer science students at the Technical University of Denmark built social bots that had a surprisingly large influence.[3]

0018-9162/16/$33.00 © 2016 IEEE

TABLE 1. Results of the DARPA Twitter Bot Challenge.

Team	Misses	Hits	Guesses	Accuracy	Speed	Final score
SentiMetrix	1	39	40	38.75	12	50.75
University of Southern California	0	39	39	39.00	6	45.00
DESPIC	7	39	46	37.25	6	43.25
IBM	4	39	43	38.00	5	43.00
Boston Fusion	9	39	48	36.75	5	41.75
Georgia Tech	56	38	94	24.00	0	24.00

The accuracy column is the value ($h - 0.25m$), where h is the number of hits (correct guesses) and m is the number of misses (incorrect guesses). The speed column equals the number of days remaining in the challenge after the team had discovered all bots. DESPIC is the Indiana University/University of Michigan team. For each team t, FinalScore(t) = Hits(t) – 0.25 × Misses(t) + Speed.

In response to this problem, DARPA held a four-week competition in February and March 2015, in which multiple teams supported by DARPA's Social Media in Strategic Communications (SMISC) program competed to identify a set of influence bots on Twitter serving as ground truth on a specific topic. Here, "ground truth" refers to specific bots that DARPA knew about. Past work regarding influence bots often has difficulty supporting claims about accuracy because there is limited ground truth (though some exceptions do exist).[4,5] With the exception of John Dickerson and his colleagues' paper,[6] no past work has looked specifically at identifying influence bots on an explicit topic. This article describes the DARPA competition and describes the methods used by the three top-ranked teams.

TWITTER BOT CHALLENGE

In the 2015 Twitter Bot Challenge, participants were asked to identify influence bots supporting a pro-vaccination discussion on Twitter. There is a vocal anti-vaccination community on the Internet and in social media.[7]

Because the challenge focused on identifying influence bots seeking to diffuse a sentiment s on a topic t, competitors had to

- separate influence bots from other types of bots;
- separate influence bots about topic t from those about other topics; and
- separate influence bots about topic t that sought to spread sentiment s from influence bots that were either neutral or opposite in sentiment.

Six teams—the University of Southern California (USC), DESPIC (Indiana University and the University of Michigan), Georgia Tech, SentiMetrix (an organization providing social media analytical solutions), IBM, and Boston Fusion—competed to discover 39 pro-vaccination influence bots. The teams did not know the actual number of bots in advance. Table 1 summarizes the final results of the competition. SentiMetrix placed first, beating the other teams by six days or more with 39 of 40 correct guesses, and USC achieved the best accuracy (39 of 39 correct guesses).

We will now examine how the top three teams (SentiMetrix, USC, and DESPIC) achieved their results.

SETUP

In fall 2014, the Pacific Social Architecting Corporation, a research group that explores how bots and technology can shape social behavior, logged records of an independent influence competition taking place on Twitter about the use of influence bots in combating misinformation online, specifically around anti-vaccine activists on Twitter. Using ground truth on the teams and the bot accounts they operated, the group subsequently developed a synthetic Twitter environment with a simulated Twitter API that played back a partially redacted set of the bots' data as part of the DARPA competition. This data consisted of

- 7,038 user accounts;
- redacted user profiles with Twitter-like format: user image, URL, number of friends and followers, plus a short user bio;
- a time-stamped tweet dataset for each user (4,095,083 tweets in all); and
- weekly network snapshots consisting of (from_user, to_user, timestamp, weight) tuples. A tuple's weight was 0 if "from_user" unfollowed "to_user," and

ADDITIONAL AUTHORS

Additional authors of this article include Andrew Stevens of SentiMetrix; Alexander Dekhtyar of California Polytechnic State University, San Luis Obispo, and SentiMetrix; Shuyang Gao, Farshad Kooti, and Yan Liu of University of Southern California; Tad Hogg of the Institute for Molecular Manufacturing; Onur Varol and Prashant Shiralkar of Indiana University; Vinod Vydiswaran and Qiaozhu Mei of the University of Michigan; and Tim Hwang of Pacific Social.

was 1 otherwise. There were 17,503 users for whom partial network data was provided—the data included user IDs that were not present in the user accounts.

Once the challenge started, teams could submit guesses to a webserver that immediately provided information on whether the guesses were correct or incorrect. Team scores were computed as follows:

- A team received 1 point for each correct guess.
- A team lost 0.25 points for each false positive (labeled "misses" in Table 1).
- A team that guessed all the bots *d* days before the challenge ended received *d* extra points.

The third scoring clause provided a bonus for speedy guessing. Identifying real-world bots early on is important in counteracting an influence campaign, especially in adversarial situations or attempts to influence an election.

SentiMetrix guessed all bots on day 16 of 28, receiving 12 bonus points. Of 40 guesses made, only 1 was wrong, for a final score of 50.75. The speed bonus gave SentiMetrix a nearly one-week edge over the immediate competitors (USC and DESPIC). Both USC and DESPIC found all bots six days later. USC had perfect precision, while DESPIC had seven erroneous guesses.

BOT DETECTION APPROACHES

The top three teams all found that machine-learning techniques alone were insufficient because of a lack of training data. However, a semi-automated process that included machine learning proved useful.

IDENTIFYING REAL-WORLD BOTS EARLY ON IS IMPORTANT IN COUNTERACTING AN INFLUENCE CAMPAIGN, ESPECIALLY IN ADVERSARIAL SITUATIONS.

All teams started with supervised learning. USC used unsupervised outlier detection in conjunction with other evidence, DESPIC and SentiMetrix used clustering algorithms in the challenge, and DESPIC also tried an online prediction strategy. None of the teams found existing Sybil detection methods useful.[8,9] All teams benefited from previous influence-bot studies.[4,6,10,11]

Creating a training set

All but one team used past work to build a profile Prof(*u*) of user *u*.[4,6] SentiMetrix's SentiBot[6] identified influence bots from the 2014 Indian election[12] over 10 months, amassing a dataset exceeding 17 million users, 25 million tweets, and 45 million edges. In addition, Kyumin Lee and his colleagues' dataset (collected from 60 social honeypots deployed over 7 months involving 42,000 users) was used by two teams to separate bots from nonbots.[4] Regardless of the method used, the features listed below were of interest to all teams.

Tweet syntax. This category considered the following:

- Whether the syntax of the user's tweets was similar to the natural language generation program ELIZA[5] and autogenerated language.[13]
- Average number of hashtags, user mentions, links, and special characters in tweets.
- Average number of retweets by the user.
- Whether the tweets are geo-enabled.
- Percentage of tweets ending with punctuation, hashtag, or link (such tweets might be automatically generated).

Tweet semantics. This category considered the following:

- Number of posts related to vaccination.
- User's average sentiment score in vaccination-related tweets.
- Measures of contradiction in posts on vaccination-related tweets using functions such as contradiction rank,[6] which measures variation between the user's sentiment across a set of topics and the sentiments of the user's neighbors on the same topics.
- Positive (or negative) sentiment strength,[6] measuring the average sentiment strength of the user's positive (or negative) tweets.
- Most frequent topics tweeted about by the user.

- Number of languages in which tweets were generated (accounts posting tweets in many languages might be bots).
- Sentiment inconsistency. Paybots often copy a link from a popular Twitter user and then replace the micro URLs from the original post with a spurious link to a site where the paybot owners are paid for generating views.[6] This feature analyzes whether sentiment in the tweet content varied significantly (in this case, on the topic of vaccination) as compared to a sentiment in a URL embedded within the tweet.

Teams used neurolinguistic programming tools such as latent Dirichlet allocation[14] for topic detection, and AVA (a sentiment analysis framework)[15] and OASYS[16] (an opinion analysis system) to assign sentiment scores in the [–1,+1] range. USC used a tripartite graph clustering approach to infer tweet-level and user-level sentiment.[17]

Temporal behavior features. This category tracked how user sentiments changed over time, including:

- Variance in tweet sentiment over time.[6] SentiMetrix was able to identify users who had an explicit infiltration strategy of posting anti-vaccination tweets to engage the anti-vaccination community and later switched to a pro-vaccination stance.
- Entropy of inter-tweet time distribution[10,19] (algorithmic tweeting should have some temporal regularities that are reflected in relatively low entropy of the corresponding distribution).
- Predictability of tweet timing based on a transfer entropy approach.[20]
- The duration of the longest session by a user without any short (5- or 10-minute) breaks (users who have a day-long session without any breaks are not likely to be human).
- Average number of tweets per day (if this number is large, it increases the probability that the user is a bot).
- Percentage of dropped followers (a user who dropped a lot of followers compared to the number of people he or she was following might be anomalous).
- Signal-to-noise ratio. DESPIC tracked the ratio of mean to standard deviation, minimum, maximum, and entropy of these values to detect abrupt changes in users' metadata (followers, followees, posts, and so on).

User profile features. This category considered the following:

- Does the user's profile have a photo? If so, is it from a stock image database?
- Does the user's profile have an associated URL? If so, does the URL have a clone elsewhere? (A URL that was a clone increased the level of suspicion.)
- Does the user name appear autogenerated? SentiMetrix generated several heuristics for such tests, for example, by comparing screen names with user names after splitting on spaces /underscores and looking for common substrings.
- Number of posts/retweets /replies/mentions.
- Number of followers/followees.
- Number of sources used, such as mobile applications, desktop browsers, or "null" for missing sources.
- GPS coordinate availability for the user's tweets.
- Similarity of the user profile to known bots (SentiMetrix measured this using Jaccard similarity, and DESPIC used cosine similarity).

Network features. Teams used network-related sources including

- Average deviation of user sentiment scores from followers and followees.
- In and out degree centrality.
- Average clustering coefficient of retweet and mention network associated with each user.
- PageRank and betweenness centrality of users in both retweet and mention networks.
- Variables related to star and clique networks associated with users. A star network has a single central node that is connected to all other nodes. In a clique network, every two vertices are linked.
- Number of known bots followed by a user (a user following several known bots is more likely to be a bot).
- Number/percentage of bots in the cluster that a user belonged to (if a clustering algorithm places the user in a cluster with many bots, the user is more likely to be a bot).

Some teams added features once the challenge started and some bots had been discovered. SentiMetrix started with 66 features (which increased to

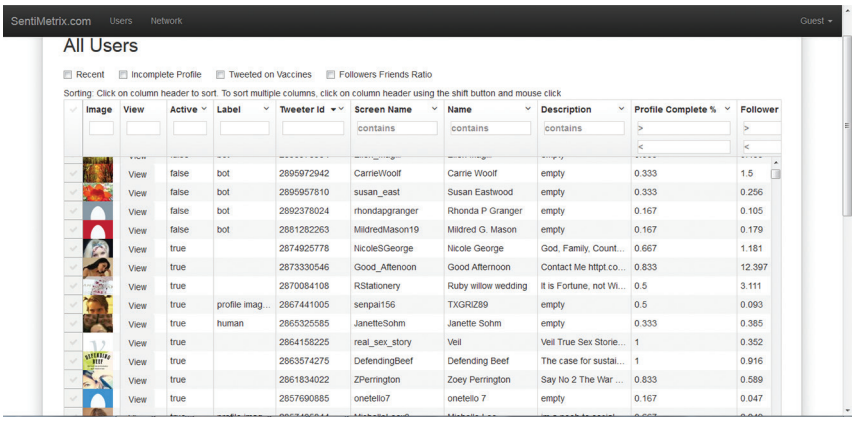

FIGURE 1. SentiMetrix's bot analysis dashboard for viewing Twitter user information.

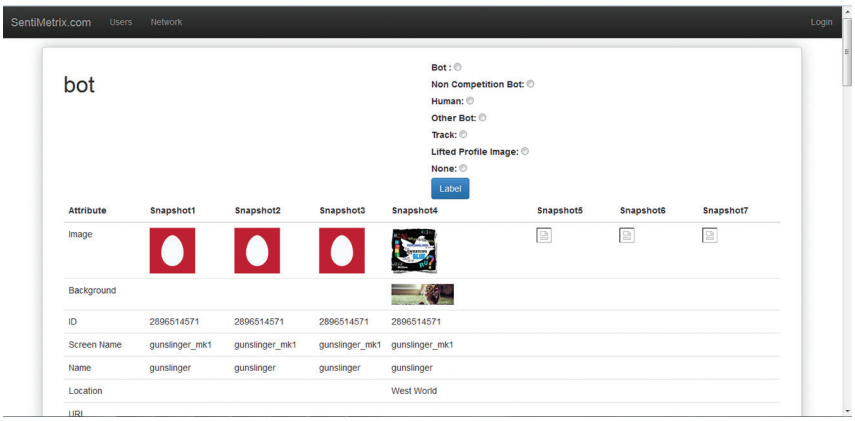

FIGURE 2. Details of a Twitter user profile on the SentiMetrix dashboard.

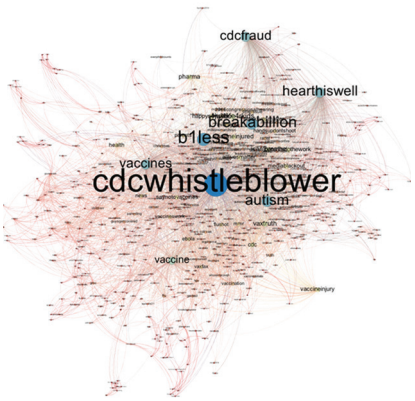

FIGURE 3. DESPIC's hashtag co-occurrence network.

175 by the end of the challenge), DESPIC used 98, and USC used 47. Teams were able to use insights from past work[4,6] to identify a small number of suspicious accounts that were then manually confirmed as bots. However, not all bots could be found using these past insights. For instance, SentiMetrix initially identified 4 bots this way and then used clustering and network analysis to identify 25 more bots. They then used support vector machines (SVMs) to predict the remaining 10 bots using the features described here and analytic tools. Similarly, USC detected the first 4 bots by combining outlier detection with content analysis and manual inspection. The next 21 bots were detected using a combination of network analysis (for example, connection to known bots), content and sentiment analysis, and semisupervised clustering of users based on the features described here.

Feature analysis

The feature data for each user was periodically updated. SentiMetrix automatically updated its feature data overnight. The top three teams used internal dashboards, allowing team members to navigate and display competition data. Teams used multiple analytical tools for competition bot prediction.

Bot analysis dashboards. All three teams used bot analysis dashboards. SentiMetrix's dashboard provided details on every user account in the DARPA challenge. Figure 1 shows the main screen, which gives the analyst a bird's-eye view of all users.

Each user has a flag displaying its label and whether the user is "active." Some users are labeled "bots" (identified as bots by the system and confirmed by human inspection), and others as "human" with a flag (for example, "profile image mismatch") that suggests that something suspicious is going on. These labels were used during clustering and SVM training, allowing SentiMetrix to discover additional groups of users.

A user summary shows how complete the user profile is and a description of the user (for example, follower/followee ratio). A number of additional variables are associated with each user (not shown in Figure 1). Many of these variables can give clues to analysts about whether a user is suspicious. The SentiMetrix dashboard allows analysts to query the profiles and to sort them in descending order of columns (top of Figure 1). It also provides information specific to each user (shown when a particular user is selected), including snapshots of the network at different times, which were updated as the competition proceeded.

Figure 2 shows the details of a Twitter user (gunslinger_mk1). This user had no profile image initially, but at some point

during the competition, a profile image was updated. Details of the background image are also shown. The top left of the screen shows that the system classified gunslinger_mk1 as a bot.

Bot analysis algorithms. The top three teams used several bot analysis algorithms, some of which are described here.

Hashtag co-occurrence network. Starting with a provided list of vaccine-related hashtags, DESPIC constructed a hashtag co-occurrence network. Nodes represent unique hashtags, and edges between two nodes are weighted by the number of times these two hashtags co-occur in a tweet (see Figure 3). DESPIC used these hashtag co-occurrence networks to identify other campaign-related hashtags, enriching the list of competition-relevant keywords. These were later used to separate users into pro- and anti-vaccine categories. The proportion of tweets containing any of these hashtags resulted in a strongly predictive feature.

Distance measures. The DESPIC team identified additional bots by computing the cosine similarity between users and known bots. Figure 4 shows the kernel density estimation of the pairwise cosine distance between pairs of feature vectors characterizing two bots, compared to bot–human pairs. The distances between bot pairs are much smaller than bot–human pairs. The bot–bot distance exhibits a bimodal distribution that reflects the presence of two types of bots designed by two separate noncompeting teams that were involved in independently creating the bots as part of Pacific Social's prior work. SentiMetrix achieved similar success using Jaccard distance.

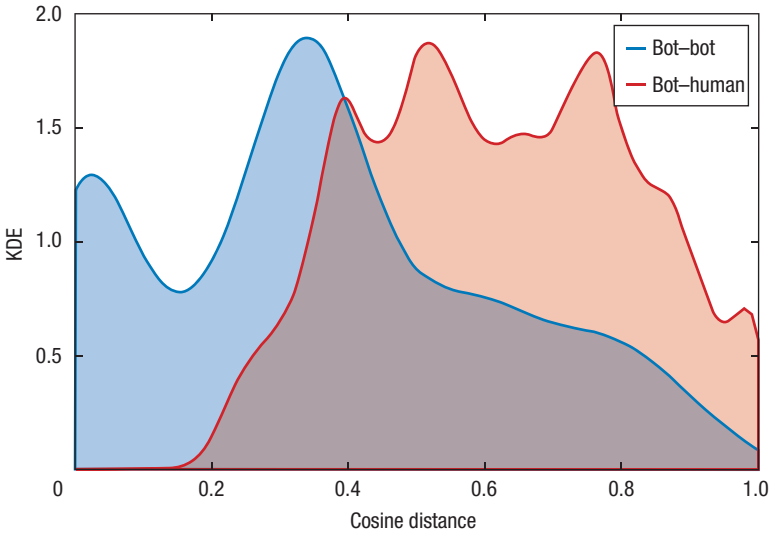

FIGURE 4. Kernel density estimation (KDE) of the cosine distance between bot–bot pairs and bot–human pairs.

Online prediction. DESPIC also adopted a multiarm bandit–based online prediction strategy. This refers to someone trying to pull the arms of multiple slot machines to maximize his payoff. In doing this, he learns the probability distribution according to which the machines provide a payoff. "Arms" were initialized with a set of binary classifiers and a hedge-like algorithm[21] that decided which arm to pull next. Each arm assigned each user account a prediction score between 0 and 1 (the higher the score, the higher the likelihood of being a bot). The hedge-like algorithm initially assigned uniform weights to all arms, and then used a multiplicative scheme to update the arm weights. After each round, it produced a final "bot score" for each user as a weighted average of the prediction scores of each arm. It then selected the account with the highest bot score as the next guess. Upon receiving the feedback score x (positive or negative), the weight of all classifiers was multiplied by a factor of $e^{x \times f_j}$, where f_j is classifier j's prediction score for that guessed account. Thus, accurate classifiers ("arms") gradually gained weight, while inaccurate ones lost weight.

Outlier detection. USC and SentiMetrix assumed it would be inefficient for bot designers to handcraft bots one at a time; they both expected that one program would generate a number of bots by varying one or more parameters in the bot-creation algorithm. Of course, a number of such bot-creation programs could be deployed by real-world bot developers. Because this would lead to similarities among bots created by the same program, two detection methods were used.

USC first applied orthogonal nonnegative matrix factorization (NMF)[22] to the data features to find a low-dimensional vector representation of each user. They then used a clustering-based outlier detection algorithm to find outliers in this low-dimension latent space. Next, they performed microlevel clustering through two approaches. In the first approach, they used the same feature representation for outliers and then reapplied NMF to cluster outliers. In the second approach, they created a similarity graph of outliers using K-nearest neighbors (KNN) search and then used modularity maximization–based community detection[23] to cluster similar outliers. USC's analysis indicates that all the confirmed bots were reported as outliers, yielding a recall score of 1.0. USC also did not make a single false positive, resulting in the only perfect accuracy score in the competition. In

contrast to USC, SentiMetrix used the well-known DBSCAN[24] algorithm to generate clusters and then prioritized which users in these clusters were likely to be bots through an analysis of their features and the similarities between those features and features of other known bots.

OVERALL FRAMEWORK FOR DETECTING INFLUENCE BOTS

Because adversaries are using different and increasingly sophisticated bot-generation methods, we believe that using machine learning alone to identify bots will be inadequate going forward. From our experience, we learned that bot detection is a semiautomated process that builds on four broad techniques: inconsistency detection and behavioral modeling, text analysis, network analysis, and machine learning. Effective bot detection requires the following carefully designed workflow, with strong supporting software:

- *Initial bot detection*. In the DARPA challenge, team members used four broad classes of cues to uncover an initial set of bots: heuristics (DESPIC looked for bots that used stock images for profile photos), behaviors (USC and SentiMetrix looked at the number of tweets posted over extended periods of time), linguistics (SentiMetrix looked for ELIZA-style tweets[5] and tweets with unusual grammar), and inconsistencies (for example, MaryJones17, a user who mostly talks about college, has a photo of a bearded older man).
- *Clustering, outliers, and network analysis*. Though only a few simple bots might be found in Step 1, they are very valuable. Bots connect to each other to inflate follower counts and to increase retweets. As most bot developers write pieces of code that vary parameters to generate bots, the shared parameters might create clusters. This means that clusters containing known bots might include other bots. In the DARPA challenge, SentiMetrix exploited these properties to find numerous bots, while USC used outlier analysis to find bots that are distant from all clusters. USC used local ego networks (subnetworks consisting of a node and its immediate neighbors) of known bots to gain insight into the bots' structural connectivity patterns, generating more candidates for guesses.
- *Classification/outlier analysis*. Once a certain number of bots and humans are detected, we can identify other bots using standard classifiers. For instance, in the DARPA challenge, once 29 bots had been found, SentiMetrix used SVMs to immediately find the remaining 10 bots.

A major problem faced by the top two teams (SentiMetrix and USC) was determining when to stop guessing. Both teams stopped when they were unable to find any more credible bots to guess. As DARPA announced that two teams had found all the bots immediately after the fact, the other teams could guess the number of bots by examining the SentiMetrix and USC scores, though we do not know if they in fact did this.

The main takeaway from the DARPA challenge is that a bot-detection system needs to be semisupervised. All teams used human judgment to augment automated bot identification processes. Interfaces that easily explain why a particular Twitter account is considered a bot are particularly important. These interfaces must include effective visualizations that highlight the top suspected accounts and explain why they are suspicious. Such interfaces must allow analysts to provide feedback and use that feedback to improve detection accuracy.

As influence bots become more sophisticated, we need to significantly enhance the tools that help analysts detect these bots. The initial bot-detection step needs to be fully automatized and to be supported by a large collection of tools that systematically cover the search space. The clustering, outliers, and network analysis stage will need to present a toolbox that detects additional bots by looking at clusters of users and outliers. Suspects detected by different algorithms will need to be merged into a single suspect list. Powerful visualization methods are needed throughout the system. Once many bots have been discovered, along with benign accounts, traditional classifiers can generate additional candidate bots.

Although the methods described here were developed for detecting fully automated bots, we believe these methods can also be used for detecting human-orchestrated influence operations, which could help counteract terrorism and political turmoil.

ACKNOWLEDGMENTS

We thank DARPA and the US Army for funding this project under contracts W911NF-12-C-0026 and W911NF-12-1-0034.

REFERENCES

1. S. Shane and B. Hubbard, "ISIS Displaying a Deft Command of Varied Media," *The New York Times*, 20 Aug. 2014; www.nytimes.com/2014/08/31/world/middleeast/isis-displaying-a-deft-command-of-varied-media.html.
2. P.R. Gregory, "Inside Putin's Campaign of Social Media Trolling and Faked Ukrainian Crimes," *Forbes*, 11 May 2014; www.forbes.com/sites/paulroderickgregory/2014/05/11/inside-putins-campaign-of-social-media-trolling-and-faked-ukrainian-crimes.
3. S. Lehmann and P. Sapieżyński, "You're Here Because of a Robot," blog, 4 Dec. 2013; http://sunelehmann.com/2013/12/04/youre-here-because-of-a-robot.
4. K. Lee, B.D. Eoff, and J. Caverlee, "Seven Months with the Devils: A Long-term Study of Content Polluters on Twitter," *Proc. AAAI Int'l Conf. Weblogs and Social Media* (ICWSM 11), 2011; www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/download/2780/3296.
5. J. Weizenbaum, "ELIZA—A Computer Program for the Study of Natural Language Communication between Man and Machine," *Comm. ACM*, vol. 9, no. 1, 1966, pp. 36–45.
6. J.P. Dickerson, V. Kagan, and V.S. Subrahmanian, "Using Sentiment to Detect Bots on Twitter: Are Humans More Opinionated Than Bots?," *Proc. IEEE/ACM Int'l Conf. Advances in Social Networks Analysis and Mining* (ASONAM 14), 2014, pp. 620–627.
7. A. Kata, "Anti-vaccine Activists, Web 2.0, and the Postmodern Paradigm–An Overview of Tactics and Tropes used Online by the Anti-vaccination Movement," *Vaccine*, vol. 30, no. 25, 2012, pp. 3778–3789.
8. G. Danezis and P. Mittal, "SybilInfer: Detecting Sybil Nodes using Social Networks," tech. report MSR-TR-2009-6, Microsoft, 2009; http://research.microsoft.com/apps/pubs/default.aspx?id=78896.
9. H. Yu et al., "Sybilguard: Defending Against Sybil Attacks via Social Networks," *IEEE/ACM Trans. Networking*, vol. 16, no. 3, 2008, pp. 576–589.
10. Z. Chu et al., "Detecting Automation of Twitter Accounts: Are You a Human, Bot, or Cyborg?" *IEEE Trans. Dependable and Secure Computing*, vol. 9, no. 6, 2012, pp. 811–824.
11. J. Ratkiewicz et al., "Truthy: Mapping the Spread of Astroturf in Microblog Streams," *Proc. 20th Int'l Conf. Companion on World Wide Web* (WWW 11), 2011, pp. 249–252.
12. V. Kagan, A. Stevens, and V.S. Subrahmanian, "Using Twitter Sentiment to Forecast the 2013 Pakistani Election and the 2014 Indian Election," *IEEE*

ABOUT THE AUTHORS

V.S. SUBRAHMANIAN is a professor of computer science, a past director of the University of Maryland Institute for Advanced Computer Studies, and a founder of SentiMetrix. Contact him at vs@cs.umd.edu.

AMOS AZARIA is a postdoctoral researcher at Carnegie Mellon University in the Machine Learning department and was part of the SentiMetrix team. Contact him at amos.azaria@gmail.com.

SKYLAR DURST is a graduate student of computer science at the California Polytechnic State University, San Luis Obispo, and a data architect at SentiMetrix. Contact him at skylar@sentimetrix.com.

VADIM KAGAN is a founder and president of SentiMetrix. Contact him at kagan@sentimetrix.com.

ARAM GALSTYAN is a project leader at the Information Sciences Institute and a research associate professor of computer science at the University of Southern California (USC). Contact him at galstyan@isi.edu.

KRISTINA LERMAN is a project leader at the Information Sciences Institute and a research associate professor in the USC Computer Science Department. Contact her at lerman@isi.edu.

LINHONG ZHU is a computer scientist at the Information Sciences Institute at USC. Contact her at linhong@isi.edu.

EMILIO FERRARA is a computer scientist at the Information Sciences Institute at USC. He was at Indiana University when this research was conducted. Contact him at emilio.ferrara@gmail.com.

ALESSANDRO FLAMMINI is an associate professor in the School of Informatics and Computing at Indiana University. Contact him at aflammin@indiana.edu.

FILIPPO MENCZER is a professor of informatics and computer science and the director of the Center for Complex Networks and Systems Research at Indiana University. Contact him at fil@indiana.edu.

Intelligent Systems, vol. 30, no. 1, 2015; http://dx.doi.org/10.1109/mis.2015.16.
13. R.F. Simmons, "Natural Language Question-Answering Systems: 1969," *Comm. ACM*, vol. 13, no. 1, pp. 15–30.
14. D.M. Blei, A.Y. Ng, and M.I. Jordan, "Latent Dirichlet Allocation," *J. Machine Learning Research*, vol. 3, 2003, pp. 993–1022.
15. V.S. Subrahmanian and D. Reforgiato, "AVA: Adjective-Verb-Adverb Combinations for Sentiment Analysis," *IEEE Intelligent Systems*, vol. 23, no. 4, 2008, pp. 43–50.
16. C. Cesarano et al., "OASYS: An Opinion Analysis System," *Proc. AAAI Spring Symp. Computational Approaches to Analyzing Weblogs*, 2004; www.umiacs.umd.edu/publications/oasys-opinion-analysis-system-0.
17. L. Zhu et al., "Tripartite Graph Clustering for Dynamic Sentiment Analysis on Social Media," *Proc. ACM SIGMOD Int'l Conf. Management of Data* (SIGMOD 14), 2014, pp. 1531–1542.
18. L. Hong and B.D. Davison, "Empirical Study of Topic Modeling in Twitter," *Proc. Workshop Social Media Analytics* (SOMA 10), 2010, pp. 80–88.
19. R. Ghosh, T. Surachawala, and K. Lerman, "Entropy-based Classification of 'Retweeting' Activity on Twitter," *Proc. Social Network Analysis Workshop* (SNA-KDD 11), 2011; http://arxiv.org/abs/1106.0346.
20. G. Ver Steeg and A. Galstyan, "Information Transfer in Social Media," *Proc. 21st Int'l Conf. World Wide Web* (WWW 12), 2012, pp. 509–518.
21. Y. Freund and R.E. Schapire, "A Decision-Theoretic Generalization of Online Learning and an Application to Boosting," *J. Computer and System Sciences*, vol. 55, no. 1, 1997, pp. 119–139.
22. D.A. Cai, "Graph Regularized Nonnegative Matrix Factorization for Data Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, 2010, pp. 1548–1560.
23. V.D. Blondel et al., "Fast Unfolding of Communities in Large Networks," *J. Statistical Mechanics: Theory and Experiment*, vol. 2008, 2008; http://dx.doi.org/10.1088/1742-5468/2008/10/p10008.
24. H.P. Kriegel and M. Pfeifle, "Density-based Clustering of Uncertain Data," *Proc. 11th ACM SIGKDD Int'l Conf. Knowledge Discovery in Data Mining* (KDD 05), 2005, pp. 672–677.
25. T.E. Nissen, "Terror.com: IS's Social Media Warfare in Syria and Iraq," *Military Studies Magazine: Contemporary Conflicts*, vol. 2, no. 1, 2014; www.fak.dk/en/news/magazine/Documents/ISSUE%2002,%20VOLUME%2002/Terror_com_ISs_Social_Media_Warfare_in_Syria_and_Iraq.pdf.