Predicting Susceptibility to Social Bots on Twitter

Randall Wald, Taghi M. Khoshgoftaar, Amri Napolitano
Florida Atlantic University
Email: {rwald1, khoshgof}@fau.edu, amrifau@gmail.com

Chris Sumner
Online Privacy Foundation
Email: chris@onlineprivacyfoundation.org

***Abstract*—The popularity of the Twitter social networking site has made it a target for social bots, which use increasingly-complex algorithms to engage users and pretend to be humans. While much research has studied how to identify such bots in the process of spam detection, little research has looked at the other side of the question — detecting users likely to be fooled by bots. In this paper, we examine a dataset consisting of 610 users who were messaged by Twitter bots, and determine which features describing these users were most helpful in predicting whether or not they would interact with the bots (through replies or following the bot). We then use six classifiers to build models for predicting whether a given user will interact with the bot, both using the selected features and using all features. We find that a users' Klout score, friends count, and followers count are most predictive of whether a user will interact with a bot, and that the Random Forest algorithm produces the best classifier, when used in conjunction with one of the better feature ranking algorithms (although poor feature ranking can actually make performance worse than no feature ranking). Overall, these results show promise for helping understand which users are most vulnerable to social bots.**

***Index Terms*—Twitter, social bots, feature selection**

I. Introduction

Twitter is one of the most popular social networks, despite — or possibly because of — its limitations. User posts are limited to 140 characters, and the privacy model is extremely limited: a whole account is either private (only sharing posts with friends) or public, and most users choose "public," sharing all of their content freely with the world. In addition, following a user is not necessarily reciprocal: because all posts are public, following a user merely subscribes a follower to their public posts, and thus users are encouraged to follow individuals they do not know personally. This has led to many celebrities using Twitter as a means of connecting with their fans, because they can update their millions of fans with a single 140-character tweet (Twitter's term for a post).

Although the open nature of Twitter can mitigate some privacy concerns (since users are aware that everything posted is public and should hopefully act accordingly), this does not mean everything is as it seems. Due to the site's popularity, and especially due to the culture of following individuals with interesting content, it has become a major target for marketing and social manipulation. Spam accounts post links to paid content, and users shill for companies while pretending to be independent fans of the company. Moreover, as users and Twitter itself become more aware of spam, automated accounts are growing more intelligent, moving beyond simple reposts of boilerplate ad content to attempt to engage with users. These social bots pretend to be human in order to gain followers and replies from their targets, and then exploit this trust to promote a product or agenda.

These social bots have been studied in the context of spam detection, so they may be filtered and removed from the site, but in order for social bots to prosper users must be fooled by them. Less work has gone into what traits make users susceptible to these bots, however. In the present work, we present a novel case study with 610 users, all of whom were contacted by a Twitter bot through an @ message. Some users interacted with the bot (either through replying directly or through following the bot), and a number of features (both demographic and linguistic) were culled from the users' profiles. We then endeavored to understand how these features contribute to whether a user will interact with a bot.

Two collections of experiments are performed, both employing a set of ten feature rankers to order the features in terms of importance. In the first experiment, the top features from the various ranked lists were compared, to discover which features were most useful in predicting an individual's vulnerable to interaction with a bot. Here, we found that a user's Klout score and total number of friends (individuals the user both followed and was followed by) were the strongest predictors, while other features indicative of social engagement also ranked highly as useful for predicting bot interaction.

In our second experiment, we used six different machine learning classifiers to build models for predicting this interaction, either using all of the features or only using those selected by the feature rankers (with six different feature subset sizes: 5, 10, 15, 20, 25, and 30). For this second experiment, we found that the Random Forest classifier gave the greatest performance, in particular when used with the Geometric Mean ranker; however, other choices of ranker actually performed worse than the Random Forest classifier applied with all features. This demonstrates the importance of choosing both a classifier (learner) and ranker which match one another and which maximize performance.

Overall, we find that user traits can be used to identify individuals more likely to interact with bots, and that this information can both be used to study how these traits directly affect vulnerability and how they may be use to build classification models. More study will be needed to analyze these connections, however.

This paper is organized as follows: Section II contains related work on the topics of bots, Twitter spam detection, and user vulnerability. Section III reviews the different techniques employed in this paper, including the classification learners, the rankers, and the performance metrics and evaluation scheme. Section IV discusses the case study and the data in more detail. We present our results in Section V, both for the feature selection alone and when using the classifiers. Finally, Section VI includes our conclusions and directions for future research.

II. Related Work

Social bots are an evolution of the chatterbot, a program which attempts to have a conversation with humans [13]. An idealized chatterbot would be able to pass the Turing test [40], meaning that a human conversing with the bot could not tell if their conversational partner was a program or a human. Research in chatterbots has advanced greatly since the original Turing test was proposed, with highlights ranging from the ELIZA bot developed in 1966 to simulate a human therapist [49] to the development of the Artificial Intelligence Markup Language (AIML) in 2001 [46] to facilitate the easy

IEEE IRI 2013, August 14-16, 2013, San Francisco, California, USA
978-1-4799-1050-2/13/$31.00 ©2013 IEEE

construction of XML-based rules for generating realistic chatterbots, including the A.L.I.C.E bot which won the Loebner Prize for artificial intelligences in 2000, 2001, and 2004 [27].

While computers have become more adept at communicating with humans, humans have found even more ways to communicate with each other. Social networks have become a major driving force in our society, and a great deal of research has focused on how such networks have evolved over time [15], [22] and how these have influenced youth culture [12], [26]. One particularly important social network is Twitter, which combines the traditional aspect of users sharing details of their lives with one another and the culture of celebrity, where famous users attract followers in order to promote their personal brands [23]. These two competing aspects have led to increased focus on Twitter influence: how much one user can affect others [2], [8], [9], and how different user networks are connected to spread information [5], [35]. While much of this research has focused on human interaction, automated agents have also begun to join the network.

As with any social network, the large number of attentive users poses an attractive target for marketers who wish to create buzz for a product. While some of this Twitter-based marketing employs actual humans who attempt to engage with the community and foster goodwill [39], the problem of spam (automatically-generated advertising messages) has become endemic, and is now a fact of life for Twitter users [17], [38]. As a result of this, more research has been conducted in detecting spam on Twitter, with techniques ranging from traditional classifiers [29], to considering the network relationships between the sender and receiver [32], [47], to evaluation of the URLs being promoted by the spammers [37]. Nonetheless, spam continues to be a problem on Twitter [16], [28]

An increasing number of organizations are using software tools to post directly to Twitter, both for legitimate and malicious (e.g., spam-generating) ends. These bots vary in sophistication, from those which simply cross-post content from an existing news site to those which use human intervention to create more realistic content [10]. While some research on these social bots has focused on other networks such as Facebook [4], [7], it has become fairly simple to construct these bots for Twitter as well [30]. Detecting these more complex bots remains a challenge [1]

To promote the study of Twitter Bots, in 2011 the Web Ecology Project began their Socialbot Challenge [20], wherein three teams competed to create bot clusters which could elicit real users to follow and reply to their bots. Each team could create as many bots as it wished, but only one would be counted for scoring, and once released the bots needed to run on their own without human intervention (although after five days, the teams were allowed to change the code for their bots to refine their strategies). The winning team employed a strategy based on many bots which simply followed the lead bot to lend it credibility, along with a dictionary of questions and responses to simulate interactivity. Over the course of the two-week challenge, this team was able to accumulate approximate 8 followers per day and 14 replies per day, with the latter metric far outweighing the other two competitors.

Although much research into Twitter bots has focused on describing and identifying the bots themselves, Wagner et al. [44] realized that the Socialbot Challenge 2011 data provided a window into the other side of the equation: the users who chose to follow bots. Targeted users were identified as "susceptible" if they interacted with one of the bots within the 14-day window of the competition, and their degree of susceptibility was based on how quickly they became "infected." Three categories of features were extracted from each user, to predict their susceptibility: 70 linguistic features (which used the Linguistics Inquiry and Word Count (LIWC) [36] package to extract word-use dimensions from users' tweets), nine network-based features using three different forms of graph generation (a follower-based directed graph, an undirected retweet-based graph, and a raw interaction graph), and 13 behavioral features based on the scope and range of tweet contents.

Using this dataset and six classification learners (Partial Least-Squares Regression, Generalized Boosted Regression, k-Nearest Neighbors, Elastic-Net Regularized Generalized Linear Models, Random Forest, and Regression Trees), the authors were able to achieve an overall accuracy of 0.71. They also found that the most important features were the user's out-degree in the interaction network (meaning they frequently retweet, follow, or otherwise interact with other users), along with other features pertaining to interaction. Other features suggest these users are more open and social than typical users, using words describing emotions and sentence structures pertaining to describing their activities. Attempts to build a more full regression model did not yield much success, however.

The present work continues the research begun by Wagner et al., exploring a new dataset with more instances (610 users, of whom 123 interacted with the bot, compared with the earlier dataset which only has 374 instances and 76 which interacted with the bot) and a wider range of classification learners from more families (focused on different forms of machine learning, rather than simply regression). In addition, the present study considers feature selection, the use of algorithms which identify the top features and enable models to be built using just these features. These facilitate both building more accurate models and identifying which traits of users puts them at risk for infection by social bots.

III. Methods

Throughout this paper, we employ six classification learners, and ten filter-based feature rankers. In this section, we describe how each of our techniques work.

A. Learners

A classification learner is an algorithm which builds a model using labeled training data (e.g., data with known class labels), and then evaluates the model using unlabeled test data (in practice, the test data's labels are known, but are only used when comparing the predicted labels with the actual labels). In this paper, six diverse learners are used: 5-Nearest Neighbor (5-NN) [33], Logistic Regression (LR) [24], Multi-Layer Perceptron (MLP) [19], Naïve Bayes (NB) [33], Random Forest with 100 trees (RF100) [6], and Support Vector Machines (SVM) [25]. Because these are each well-understood techniques, we provide only a brief discussion of how they predict class labels; an interested reader may consult the references for further information. All models were built using the Weka Machine Learning Toolkit [18], using the default parameter values unless otherwise noted.

5-Nearest Neighbor classifies instances by finding the five closest instances to the test instance and comparing the total weight of the instances from each class (using 1/Distance as the weighting factor). Logistic Regression builds a simple logistic model using all of the features in order to predict the class variable. Multilayer Perceptron builds an artificial neural network with three nodes in its single hidden layer, and 10% of the data held aside in order to validate when to stop the back-propagation procedure. Naïve Bayes uses Bayes' Theorem to determine the posterior probability of membership in a given class based on the values of the various features, assuming that all of the

features are independent of one another. Random Forest builds a set of unpruned decision trees (100 trees in this study) with each tree using only a randomly-chosen subset of the original features and a randomly-bootstrapped copy of the data for model-building, and then the final model classifies an instance based on the majority vote of the decision trees. Finally, Support Vector Machines finds a maximal-margin hyperplane which cuts through the space of instances (such that instances on one side are in one class and those on the other side are in the other class), choosing the plane which preserves the greatest distance between each of the classes. In this paper, for SVM the complexity constant "c" was set to 5.0 and the "buildLogisticModels" parameter was set to "true."

B. Feature Selection

Due to the large number of features in this dataset, we experiment using filter-based feature rankers, which rank features based on their relevance to the class attribute. These are called filter-based because they do not employ a classifier (as wrapper-based methods do), and are rankers because they rate features independently (rather than in groups as with subset evaluation). The ten feature rankers studied come from three broad groupings: three commonly-used feature selection methods (Chi-Squared [50], Information Gain [50], and ReliefF [21]), five threshold-based feature selection (TBFS) methods (Deviance [43], Geometric Mean [43], Mutual Information [3], Area Under the ROC (Receiver Operating Characteristic) Curve [11], Area Under the Precision-Recall Curve (PRC) [31]), and two first-order-statistics based methods (Signal-To-Noise [48] and Significance Analysis of Microarrays [41]). All of these techniques will be briefly described below.

1) Commonly-used feature ranking techniques: The first three feature selection techniques were chosen due to being commonly-used in the data mining and machine learning literature. Chi-Squared is a metric based on the χ^2 distribution, which is how the feature and class values would be distributed if there were no correlation whatsoever between the two. How far the actual distribution is from the theoretical no-correlation distribution shows how well the feature is correlated with the class. Information Gain is an entropy-based performance metric, based on the amount of entropy present in the partitioning of the instances based on their class values. The amount by which this entropy is reduced when the instances are first partitioned according to their feature values is how much information is gained when using that feature. ReliefF is an instance-based performance metric based on the idea of picking a random instance and comparing its feature values with those from its nearest hit (the closest instance in the same class) and its nearest miss (the closest instance in the same class). Features increase their score by being close in value in the nearest hit, but are penalized for being close in value to the nearest miss.

2) TBFS-based feature ranking techniques: Threshold-based feature ranking is a class of feature ranking techniques recently developed by our research group [14], [42]. The premise of TBFS is to consider only the two-attribute dataset consisting of the feature being examined and the class variable (which must be a binary class variable). The feature being studied is then normalized so its values lie between 0 and 1. This normalized feature value is then treated as a posterior probability, and is used to "predict" the class (according to two different rules, depending on whether higher values are associated with one class or the other). These predictions are evaluated using different classifier performance metrics for different threshold values (cutoff points where the posterior is interpreted as one class or the other), and the threshold (and direction) which optimizes the performance metric is used to calculate the quality of the feature in question. Note that although classifier performance metrics are used to evaluate the features, no actual classifiers are built; only the normalized feature values are examined. The insight is that highly-predictive features should have values which correlate with the class in much the same way that a high-performance classifier's posterior probabilities will; thus, the same performance metrics may be used. Note also that additional classifier performance metrics could be used with the TBFS framework.

As noted, five different performance metrics were employed in conjunction with the TBFS technique. Deviance is based on the minimum residual sums (sum of squared errors) found in the partitioning based on the threshold t. Because it is a measure of error, lower values of Deviance are better. Geometric Mean is the square root of the product of the true positive rate (number of true positives / total number of positive instances) and the true negative rate (number of true negatives / total number of negative instances). Mutual Information finds the mutual dependence of the two variables in question (the feature and the class variable), showing how much information about one can be used to reduce the uncertainty of the other. Area under the ROC (Receiver Operating Characteristic) Curve is a measure of the total area under the ROC curve; this curve plots the trade-off between the true positive rate and the false positive rate, for all values of the threshold t. It can be used to give a balanced view of how these two factors relate without picking a single threshold level. Finally, PRC (Area Under the Precision-Recall Curve) is similar, but considers the trade-off between precision (number of true positives / total number of instances predicted to be positive) and recall (the same as true positive rate).

3) First-order-statistics based feature ranking techniques: The final two feature ranking techniques are called "first-order-statistics based" because they both rely on first-order statistics such as mean and standard deviation. Signal-To-Noise is based on the ratio of the signal versus the noise. In particular, it is the ratio of the difference between the feature's mean values for each class over the sum of the feature's standard deviations on each class. Significance Analysis of Microarrays (SAM) uses an attribute-specific t-test for each feature to measure the strength of the correlation between each independent feature and the class attribute. Specifically, the difference in the feature's mean values in the two classes is divided by the sum of the overall standard deviation and an exchangability constant which helps prevent features with small standard deviations from having an abnormally large SAM score. Although SAM was developed for the domain of bioinformatics, it is a general technique which may be applied to any application domain.

4) Feature Subset Sizes: Following feature ranking, for our classification experiments we created feature subsets of varying sizes by choosing the top N values from each ranked list. In our experiments, N values included 5, 10, 15, 20, 25, and 30. These values were chosen to give good coverage, and because preliminary experiments demonstrated diminishing returns for larger feature subset sizes.

C. Performance Metrics and Cross-Validation

To evaluate the results of our models, we used the AUC performance metric. This value, the area under the ROC curve, is calculated just as the ROC feature ranker is (as discussed in Section III-B) The distinction is that when used as a ranker, the values of the feature to be ranked are considered as the posterior probabilities for determining true positives and false positives. When AUC is used as a performance metric, however, the actual posterior probability from the classifier is

used. To make these more distinct, in this paper we use ROC to refer to the ranker and AUC to refer to the overall performance metric.

When evaluating models, cross-validation was used. This is a process which divides the data into N equal-size subsets (folds), builds the model on $N-1$ of these, and tests the model on the Nth fold, called the hold-out fold. This process is repeated N times, so that each fold is used as the hold-out fold exactly once. The results are combined across all the N runs to define the performance metrics. In this experiment, we used ten-fold cross-validation (e.g., $N = 10$). In addition, we performed the entire cross-validation process a total of five times, and all results presented are the average across these five runs of ten-fold cross-validation.

IV. Case Study

The case study for this experiment grew out of work performed by the Online Privacy Foundation[1], an organization dedicated to understanding how users interact with social networks and the privacy implications of their actions. In a previous experiment, called the Twitter Big Five Experiment [34], [45], a number of users were solicited to take an online personality survey, rating these users according to the Big Five personality index (Agreeableness, Conscientiousness, Extroversion, Openness, Neuroticism) and the Dark Triad of negative personality traits (Narcissism, Machiavellianism, and Psychopathy). In addition, two classes of features were extracted from each individual's profile: demographic features and linguistic features. Demographic features consisted of numeric information which could be directly extracted from the profile, or generated automatically using this numeric content: this includes facts like the number of friends and followers, the number of statuses posted by the user, the length of their self-description, and so on. One demographic feature of special note is an individual's Klout score. This is a measure created by an independent company, Klout.com, to evaluate a person's overall reach in terms of social network connections. In the original Twitter Big Five Experiment, 20 demographic features were used, but for the present social bot experiment, there were 22.

The remaining features were extracted using the Linguistic Inquiry and Word Count package [36], which divides words into many different linguistic categories to represent different types of language use, and then counts how often a given individual uses words in each category. In addition, certain forms of punctuation and part of speech are counted to evaluate how these reflect on a user's writing style. These counts were collected across all of a user's tweets, to generate an overall picture of their personal writing style. A total of 70 linguistic features were extracted for each user.

Following the Twitter Big Five Experiment, a second experiment was performed on the same users, to study their response to social bots. The original pool of subjects was reduced to 610 users, who were then divided into two groups to be studied separately (although the procedure for both was identical, and as such these groups are pooled for the present work). For both groups, a Twitter bot was created which performed two tasks: it would post tweets meant to be representative of what normal Twitter users would post, and it would ask specific questions of the users in the experimental group, using Twitter's @ syntax to send these messages directly at the target users. A user was considered to have interacted with the bot if they replied to this message or if they subsequently chose to follow the bot. Only users who were part of the original Twitter Big Five Experiment were considered, even if other users followed (or sent messages to) the bot of their own accord. Thus, for each user, the 102 features were paired with a single binary class variable: whether or not that user interacted with the bot. The goal of our experiments was to understand how these features could be used to predict this interaction, both on their own and when used to augment a classification learner for building a prediction model.

[1]https://www.onlineprivacyfoundation.org/

V. Results

Two collections of experiments were conducted in this research: first, the ten feature rankers were used on the full dataset to discover the attributes most directly related to the question of predicting an individual's likelihood to interact with a social bot, and second, we built classification models (both with and without feature selection) to actually perform this prediction. These results are presented in the following two sections.

A. Feature Ranking

Table I presents a list of all features which are within the top four of any of the ranked lists (produced by the ten feature rankers), along with where each of those features was on the respective lists. We chose to limit ourselves to the top four from each ranked feature list to reduce the scope of this table: increasing to larger numbers would have added more features which appear in only a handful of lists without affecting the conclusions for those which appear in many lists. Note that all features will eventually appear on all lists, so a value of "-" only means that the given feature was not within the top four when using that feature ranker. In this table, the features themselves are sorted based on how many different lists they appeared upon (which is itself presented in the final column), and within equal-count features, by how close to the top each feature appeared in its respective lists.

These thirteen features represent the following traits of each user:

a) kloutscore: This is a metric calculated by the private company Klout.com, which collects information from a user's Facebook, Twitter, G+, LinkedIn, and other social networking profiles to determine their overall social influence.

b) friends_count: This is the number of individuals who the user in question follows (the user's out-degree).

c) followers_count: This is the number of individuals who are following the user (the user's in-degree).

d) sexual: This is an LIWC feature which counts the number of sexual references in the user's tweets.

e) Parenth: This LIWC feature counts the number of parentheses in the user's tweets.

f) notifications: This reflects whether or not the user has notifications enabled.

g) Percent_FF: This is the percentage of tweets the user posted including the phrase "Follow Friday" or the corresponding hashtag (#FF), representing participation in Twitter's "Follow Friday" events.

h) log_status: This is the natural logarithm of the total number of statuses.

i) WC: This is the overall word count of all of the user's tweets.

j) geo_enabled: This is whether or not the user has chosen to add their location to tweets.

k) Desc10: This is a binary feature (hence 1 or 0) reflecting whether or not the user has posted a description of themselves.

l) Comma: This is the LIWC feature reflecting how often commas are found in the user's tweets.

Feature	Feature Ranking Technique										Total
	CS	IG	RF	Dev	GM	MI	ROC	PRC	S2N	SAM	Lists
kloutscore	2	2	-	1	1	1	1	-	1	1	8
friends_count	1	1	-	-	2	2	2	1	3	-	7
followers_count	3	3	-	-	3	4	3	2	-	-	6
sexual	-	-	-	3	-	-	4	-	4	3	4
Parenth	4	4	-	4	-	-	-	-	-	-	3
notifications	-	-	2	-	-	-	-	-	-	2	2
Percent_FF	-	-	-	2	-	3	-	-	-	-	2
log_status	-	-	-	-	-	-	-	3	2	-	2
WC	-	-	4	-	-	-	-	-	-	4	2
geo_enabled	-	-	1	-	-	-	-	-	-	-	1
Desc10	-	-	3	-	-	-	-	-	-	-	1
Comma	-	-	-	-	4	-	-	-	-	-	1
statuses_count	-	-	-	-	-	-	-	4	-	-	1

TABLE I: Placement of features within top 4 of each ranked list

m) statuses_count: This is a raw count (without using the natural logarithm function) of the user's statuses.

Overall, we see that the kloutscore and number of friends are the strongest predictors of whether or not an individual will interact with a social bot: these appear within the top four features of eight and seven (respectively) of our ten feature ranking techniques, with kloutscore being the top feature six times and number of friends being the top feature three times. It makes sense that these two features are at the top, because they both reflect involvement with social networking in general: through Klout.com's calculation of social influence, and through the number of individuals who are followed by the user in question. Individuals who are highly engaged in social networking would seem to be more likely to interact with an unknown user (such as a social bot) even if this user might have imperfect grammar or word use.

It is interesting to note, however, that despite these features appearing on top so often, the RF ranker does not score either of these features within its top four, while PRC chooses friends_count but not kloutscore and both Dev and SAM choose kloutscore but not friends_count. This demonstrates that even when a feature seems to have strong correlation with the class in question, not all feature ranking techniques will agree on it.

Additional features which are predictive of interaction with social bots include followers_count, sexual, and Parenth. The first of these goes along with the same ideas as kloutscore and friends_count: individuals who are more engaged in social networking are more likely to interact with social bots. The latter two are more interesting, however. For the sexual feature, we found that users who use more sexual language and terminology are more likely to interact with social bots. On the other hand, users who used more parenthesis in their writing were less likely to interact with bots. This may relate to the concept of openness and forthright speaking, as opposed to complex sentence constructions which employ parenthesis. Individuals who are more open to new experiences (and who are perhaps less likely to use complex sentence structures which bots find difficult to replicate) are perhaps more interested in interacting with an "individual" who is actually a bot.

Overall, we see that a wide range of features are important for predicting interaction with social bots, but that some features are more important than others. This justifies the idea that classification models (which make predictions based on many features) will give more useful results than statistical correlations which only consider a single feature paired with the class value, and that feature selection can help improve performance by removing those features which aren't contributing to the model.

B. Classification

Our classification experiments consisted of using our six learners (5-NN, LR, MLP, NB, RF100, and SVM) both on their own (without feature selection) and in conjunction with the ten feature ranking algorithms presented earlier (CS, IG, RF, Dev, GM, MI, ROC, PRC, S2N, and SAM). In all cases, we performed five runs of ten-fold cross-validation. For the experiments using feature selection, the feature ranker was applied to the nine training folds, the top features were selected (with subset sizes of 5, 10, 15, 20, 25, and 30), and these features were used to build a model from the training folds which would be evaluated on the test fold. Errors on the test folds were collected together to create a single performance metric for the entire run of cross-validation, rather than simply averaging the results across the test folds. Overall, with five runs, six learners, ten rankers, and six feature subset sizes, we have 30 results using no feature selection and 1800 results which use feature selection. All results are presented using the AUC performance metric.

Due to the number of results, especially for when using feature selection, we present these in two ways: averaged over all the learners (showing each ranker separately), and averaged over all rankers (showing each learner separately). In both cases the results with no feature selection are presented at the end, in the "No FS" column. Table II shows the results for each learner (e.g., averaged across all rankers), while Table III shows the results for each ranker (e.g., averaged across all learners). In both tables, the best performance for a given feature subset size is printed in **bold**, while the worst results for that subset size are in *italics*. Note that for Table III, the final column only contains a single value because without feature selection, the choice of ranker is meaningless. This value is the average performance across all of the no-feature-selection models.

One important note for Tables II and III is that due to computational constraints, we were unable to build a model which used the RF100 learner, the RF ranker, and feature subset size 5. Thus, all averages do not include this combination, which will slightly change the value for the RF100-subset size 5 combination in Table II and the RF-subset size 5 combination in Table III. We feel that this limitation does not meaningfully affect our conclusions.

Looking at the results in terms of the learners (Table II), we see that RF100 is the best learner across the board, for all feature subset sizes as well as for no feature selection. 5-NN, on the other hand, is the worst learner for all subset sizes aside from feature subset size 15 (where NB is the worst learner). LR is the second-best learner when feature selection is employed, although SVM is second-best without feature selection (despite usually being fourth-best with feature selection). The effectiveness of RF100, the only ensemble-

Choice of Learner	Feature Subset Size						No FS
	5	10	15	20	25	30	
5-NN	*0.55644*	*0.57336*	0.57924	*0.58051*	*0.58549*	*0.58941*	*0.56801*
LR	0.60660	0.62565	0.63600	0.64430	0.64994	0.65262	0.66909
MLP	0.59115	0.60779	0.61376	0.62031	0.62590	0.62414	0.58508
NB	0.57791	0.57755	*0.57647*	0.58432	0.59216	0.59445	0.57591
RF100	**0.60764**	**0.63320**	**0.64782**	**0.65795**	**0.66362**	**0.66811**	**0.68028**
SVM	0.57282	0.58459	0.59542	0.60172	0.60765	0.61418	0.67697

TABLE II: AUC Averaged Across All Choices of Ranker

Choice of Ranker	Feature Subset Size						No FS
	5	10	15	20	25	30	
CS	0.57887	0.58971	0.58959	0.60485	0.60830	0.61414	0.62589
IG	0.57987	0.59018	0.59088	0.60362	0.60918	0.61488	
RF	*0.54936*	*0.56670*	*0.58134*	*0.58863*	0.60217	0.60733	
Dev	0.59418	0.60978	0.61343	0.61612	0.62163	0.62641	
GM	0.58977	0.60717	0.61993	**0.63551**	**0.64375**	**0.64367**	
MI	0.59168	0.61758	0.61742	0.62055	0.62544	0.62786	
ROC	0.58876	0.61664	**0.63354**	0.62831	0.63406	0.63620	
PRC	0.58170	0.58773	0.60822	0.62386	0.62817	0.63073	
S2N	0.59160	**0.62455**	0.62789	0.63156	0.63961	0.63636	
SAM	**0.59875**	0.59354	0.59895	0.59550	*0.59563*	*0.60061*	

TABLE III: AUC Averaged Across All Choices of Learner

Choice of Ranker	Feature Subset Size						No FS
	5	10	15	20	25	30	
CS	0.61618	0.64053	0.65066	0.66152	0.66356	0.65854	0.68028
IG	0.61818	0.64107	0.65355	0.65300	0.66381	0.66087	
RF	-	*0.59460*	*0.61726*	*0.62893*	0.63110	0.64017	
Dev	0.58567	0.60364	0.62150	0.62915	*0.62927*	*0.63583*	
GM	0.62319	0.66123	**0.67211**	**0.69551**	**0.69754**	**0.70272**	
MI	**0.62683**	**0.67447**	0.66208	0.66943	0.68477	0.68204	
ROC	0.61318	0.64032	0.66812	0.68208	0.67818	0.68370	
PRC	*0.58234*	0.62745	0.65044	0.67464	0.68143	0.68766	
S2N	0.60598	0.65202	0.66413	0.65516	0.67064	0.67337	
SAM	0.59718	0.59665	0.61833	0.63008	0.63587	0.65614	

TABLE IV: AUC: Only for RF100 Learner

based learner used in this study, demonstrates why ensembles are important, especially since it performs well both with and without feature selection, a trait no other learner shares.

From this table, we also see that generally, increasing the number of features selected improves the performance of the classification models. This is true for all six learners across all six feature subset sizes tested. Notably, however, some learners perform better with feature selection than they do when using all features (5-NN, MLP, and NB), while others perform best with no feature selection at all (LR, RF100, and SVM). Specifically, the best learners were the ones which showed decreased performance with feature selection. This might suggest that feature selection cannot be used to improve performance in this application domain, but further analysis of the individual feature rankers shows a different story.

Table III presents the results for each feature ranker individually, averaged across all six learners. We see a wide range of performance here, with no ranker clearly showing superior performance in all cases: GM is best for the larger feature subset sizes (20 and above), but SAM, S2N, and ROC were best when using 5, 10, and 15 features, respectively. Overall, RF was the worst ranker in general, being on the bottom for all subset sizes other than 25 and 30 (where SAM showed worse performance). It is especially interesting to note that the average performance without feature selection falls in the middle of the various models which use feature selection (and feature subset size 30, as we again see that increased feature subset size improves performance). This shows that for some rankers, using feature selection improves performance, while for other rankers it reduces it. This can partially explain the results we saw in Table II: for each learner, some rankers will improve performance while others will reduce it, and thus the overall question of whether feature ranking will help that learner depends on the balance of these two groups of learners.

Because RF100 is the top learner overall, we wished to examine it more closely to see how the choice of feature ranker affects its performance. Table IV presents the results from this learner alone, without averaging over any other learners. We see that as in Table III, for the larger feature subset sizes GM is the best ranker, but here MI (which never previously showed superlative performance) gives the best models for feature subset sizes 5 and 10. RF and Dev show the same performance patterns as before, however, with RF being the worst for subset sizes 20 and below, while Dev is worst for sizes 25 and 30. (Due to the missing results for RF at subset size 5, we cannot be certain if the results here are worse than those for PRC at that subset size.) The most significant observation from this table, however, is that with the right choice of feature ranker, using feature ranking will improve (or at least not harm) performance: GM gives an AUC result over 0.02 better than no feature selection, while the MI, PRC, and ROC rankers are also able to improve results.

Thus, the misleading results from Table II are explained: with the right choice of feature ranking algorithm, a model using a reduced feature subset can give equal or greater performance than a model which employed all features, while both using less computational

time (due to employing fewer than 30% of the original features) and also helping practitioners understand which features are most important for the problem at hand.

VI. Conclusion

Social bots are becoming more common on sites such as Twitter, and are used to influence opinions both politically and commercially. While much research has considered how to identify these bots and remove them from the site, less work has examined the other side of the question: what makes a user likely to fall for a bot and be influenced by it. In this paper, we analyzed a dataset consisting of 610 Twitter users, all of whom were contacted by a social bot. From each user we extracted both demographic and linguistic features, and the goal of this research was to understand how these features related to a user's vulnerability to bots: that is, whether they chose to reply to or follow the bot.

We performed two related experiments to better understand this connection. In the first experiment, we performed feature ranking using ten feature ranking algorithms (Chi-Squared, Information Gain, ReliefF, Deviance, Geometric Mean, Mutual Information, Area Under the ROC Curve, Area Under the PRC Curve, Signal-To-Noise, and Significance Analysis of Microarrays) and examined those features which were consistently at the top of the list (specifically, within the top four features). We found that Klout score, total number of friends, and total number of followers are the strongest predictors of whether an individual will interact with a bot, appearing in seven (or in the case of number of followers, six) of the ten ranked lists. Beyond these features, the remaining features which appeared in two or more lists all represented individuals who are more connected with social media, with a higher number of posts, notifications, and description lengths. These also included two of the linguistic features, which demonstrated that users were more likely to interact with bots if they use a greater amount of sexual language but a lesser amount of parenthesis. Overall, this paints a picture of a user who is engaged with Twitter and is more open about their experiences, while perhaps not using certain linguistic constructs which employ parenthesis.

In our second experiment, we used six classification learners (5-Nearest Neighbor, Logistic Regression, Multi-Layer Perceptron, Naïve Bayes, Random Forest with 100 trees, and Support Vector Machines) to build models of the users in order to predict whether or not they would interact with the bot. These models were built both with and without feature selection, and when feature selection was employed the above ten rankers were used with six feature subset sizes (5, 10, 15, 20, 25, 30). Here, we found that RF100, the only ensemble learner, performed best both with and without feature selection, giving AUC performance values of 0.70272 (with the best feature ranker) and 0.68028 respectively, with SVM being second-place without feature selection (AUC 0.67697) and MLP being second-place with feature selection (AUC 0.65338, averaged across all rankers). In both cases, 5-NN produced the worst models. Comparing with and without feature selection for a single learner, the results varied strongly depending on the choice of feature selection algorithm: a good choice could improve AUC performance by as much as 0.02, while a bad choice could reduce it by at least as much. Finally, overall we found that larger feature subset sizes performed better, with size 30 often performing better than any smaller sizes. However, because using all features produced worse results for some rankers, we know that some amount of feature selection is improving results.

Overall, we find that the features we evaluated in this study can help explain which users will choose to interact with social bots. Some features make sense on their own and paint a picture of the type of user who will most likely interact with a bot, while using more features together can help build a model which will predict user interaction. This research will help users and security experts alike understand who is most vulnerable and perhaps most in need of additional aid for dealing with social bots.

Future research can continue in a number of directions. First of all, larger feature subset sizes can be evaluated, to determine the ideal number of features for building a model. In addition, the linguistic features in this study were evaluated across all of a user's tweets, regardless of whether they were general status messages, personal messages to other users, or retweets; future research could consider these separately. Finally, because this research was conducted as a follow-up of the Twitter Big Five Experiment, the personality types of these users is known, and further work could consider how these relate to an individual's likelihood of interacting with a social bot.

References

[1] A. A. Amleshwaram, N. Reddy, S. Yadav, G. Gu, and C. Yang, "CATS: Characterizing automation of Twitter spammers," in *Fifth International Conference on Communication Systems and Networks (COMSNETS)*. IEEE, 2013, pp. 1–10.

[2] E. Bakshy, J. M. Hofman, W. A. Mason, and D. J. Watts, "Everyone's an influencer: quantifying influence on twitter," in *Proceedings of the fourth ACM international conference on Web search and data mining*, ser. WSDM '11. New York, NY, USA: ACM, 2011, pp. 65–74. [Online]. Available: http://doi.acm.org/10.1145/1935826.1935845

[3] R. Battiti, "Using mutual information for selecting features in supervised neural net learning," *IEEE Transactions On Neural Networks*, pp. 537–550, 1994.

[4] Y. Boshmaf, I. Muslukhov, K. Beznosov, and M. Ripeanu, "The socialbot network: when bots socialize for fame and money," in *Proceedings of the 27th Annual Computer Security Applications Conference*, ser. ACSAC '11. New York, NY, USA: ACM, 2011, pp. 93–102. [Online]. Available: http://doi.acm.org/10.1145/2076732.2076746

[5] D. Boyd, S. Golder, and G. Lotan, "Tweet, tweet, retweet: Conversational aspects of retweeting on Twitter," in *43rd Hawaii International Conference on System Sciences (HICSS)*. IEEE, 2010, pp. 1–10.

[6] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[7] G. Brown, T. Howe, M. Ihbe, A. Prakash, and K. Borders, "Social networks and context-aware spam," in *Proceedings of the 2008 ACM conference on Computer supported cooperative work*, ser. CSCW '08. New York, NY, USA: ACM, 2008, pp. 403–412. [Online]. Available: http://doi.acm.org/10.1145/1460563.1460628

[8] J. Campo-Ávila, N. Moreno-Vergara, and M. Trella-López, "Analizying factors to increase the influence of a twitter user," in *Highlights in Practical Applications of Agents and Multiagent Systems*, ser. Advances in Intelligent and Soft Computing, J. B. Pérez, J. M. Corchado, M. N. Moreno, V. Julián, P. Mathieu, J. Canada-Bago, A. Ortega, and A. F. Caballero, Eds. Springer Berlin Heidelberg, 2011, vol. 89, pp. 69–76. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-19917-2_9

[9] M. Cha, H. Haddadi, F. Benevenuto, and K. P. Gummadi, "Measuring user influence in Twitter: The million follower fallacy," in *Proceedings of International AAAI Conference on Weblogs and Social*, 2010.

[10] Z. Chu, S. Gianvecchio, H. Wang, and S. Jajodia, "Detecting automation of Twitter accounts: Are you a human, bot, or cyborg?" *IEEE Transactions on Dependable and Secure Computing*, pp. 811–824, 2012.

[11] W. J. Conover, *Practical Nonparametric Studies*. John Wiley and Sons, 2nd edition, 1971.

[12] J. Cotterell, *Social Networks in Youth and Adolescence*, ser. Adolescence and Society. Taylor & Francis, 2007. [Online]. Available: http://books.google.com/books?id=_oK2bVXb4DoC

[13] O. V. Deryugina, "Chatterbots," *Scientific and Technical Information Processing*, vol. 37, no. 2, pp. 143–147, 2010. [Online]. Available: http://dx.doi.org/10.3103/S0147688210020097

[14] D. J. Dittman, T. M. Khoshgoftaar, R. Wald, and J. Van Hulse, "Comparative analysis of DNA microarray data through the use of feature selection techniques," in *Ninth IEEE International Conference*

on Machine Learning and Applications (ICMLA), December 2010, pp. 147–152.

[15] P. Doreian and F. Stokman, *Evolution of social networks*, ser. The journal of mathematical sociology. Gordon & Breach Publishing Group, 1997, no. v. 1. [Online]. Available: http://books.google.com/books?id=9o3vLXNTGxsC

[16] S. Ghosh, B. Viswanath, F. Kooti, N. K. Sharma, G. Korlam, F. Benevenuto, N. Ganguly, and K. P. Gummadi, "Understanding and combating link farming in the twitter social network," in *Proceedings of the 21st international conference on World Wide Web*, ser. WWW '12. New York, NY, USA: ACM, 2012, pp. 61–70. [Online]. Available: http://doi.acm.org/10.1145/2187836.2187846

[17] C. Grier, K. Thomas, V. Paxson, and M. Zhang, "@spam: the underground on 140 characters or less," in *Proceedings of the 17th ACM conference on Computer and communications security*, ser. CCS '10. New York, NY, USA: ACM, 2010, pp. 27–37. [Online]. Available: http://doi.acm.org/10.1145/1866307.1866311

[18] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[19] S. Haykin, *Neural Networks: A Comprehensive Foundation 2nd edition*. Prentice Hall, 1998.

[20] T. Hwang. (2011) Help robots take over the internet: The socialbots 2011 competition : Web ecology project. [Online]. Available: http://www.webecologyproject.org/2011/01/help-robots-take-over-the-internet-the-socialbots-2011-competition/

[21] I. Kononenko, "Estimating attributes: Analysis and extensions of relief," in *Machine Learning: ECML-94*, ser. Lecture Notes in Computer Science, F. Bergadano and L. De Raedt, Eds. Springer Berlin / Heidelberg, 1994, vol. 784, pp. 171–182. [Online]. Available: http://dx.doi.org/10.1007/3-540-57868-4_57

[22] R. Kumar, J. Novak, and A. Tomkins, "Structure and evolution of online social networks," in *Link Mining: Models, Algorithms, and Applications*, P. S. Yu, J. Han, and C. Faloutsos, Eds. Springer New York, 2010, pp. 337–357. [Online]. Available: http://dx.doi.org/10.1007/978-1-4419-6515-8_13

[23] H. Kwak, C. Lee, H. Park, and S. Moon, "What is Twitter, a social network or a news media?" in *Proceedings of the 19th international conference on World wide web*, ser. WWW '10. New York, NY, USA: ACM, 2010, pp. 591–600. [Online]. Available: http://doi.acm.org/10.1145/1772690.1772751

[24] S. Le Cessie and J. C. V. Houwelingen, "Ridge estimators in logistic regression," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, pp. 191–201, 1992.

[25] T.-Y. Liu, "EasyEnsemble and feature selection for imbalance data sets," in *IJCBS '09: International Joint Conference on Bioinformatics, Systems Biology and Intelligent Computing, 2009.*, August 2009, pp. 517–520.

[26] S. Livingstone and D. R. Brake, "On the rapid rise of social networking sites: New findings and policy implications," *Children & Society*, vol. 24, no. 1, pp. 75–83, 2010. [Online]. Available: http://dx.doi.org/10.1111/j.1099-0860.2009.00243.x

[27] H. Loebner. (2013) Home page of the Loebner Prize. [Online]. Available: http://www.loebner.net/Prizef/loebner-prize.html

[28] C. Lumezanu and N. Feamster, "Observing common spam in Twitter and email," in *Proceedings of the 2012 ACM conference on Internet measurement conference*, ser. IMC '12. New York, NY, USA: ACM, 2012, pp. 461–466. [Online]. Available: http://doi.acm.org/10.1145/2398776.2398824

[29] M. McCord and M. Chuah, "Spam detection on Twitter using traditional classifiers," in *Autonomic and Trusted Computing*, ser. Lecture Notes in Computer Science, J. M. A. Calero, L. T. Yang, F. G. Mármol, L. J. García Villalba, A. X. Li, and Y. Wang, Eds. Springer Berlin Heidelberg, 2011, vol. 6906, pp. 175–186. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23496-5_13

[30] S. M. Rodrigo and J. G. F. Abraham, "Development and implementation of a chat bot in a social network," in *Ninth International Conference on Information Technology: New Generations (ITNG)*. IEEE, 2012, pp. 751–755.

[31] N. Seliya, T. M. Khoshgoftaar, and J. Van Hulse, "A study on the relationships of classifier performance metrics," in *21st International Conference on Tools with Artificial Intelligence*, November 2009, pp. 59–66.

[32] J. Song, S. Lee, and J. Kim, "Spam filtering in Twitter using sender-receiver relationship," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, R. Sommer, D. Balzarotti, and G. Maier, Eds. Springer Berlin Heidelberg, 2011, vol. 6961, pp. 301–317. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-23644-0_16

[33] J. Souza, N. Japkowicz, and S. Matwin, "Stochfs: A framework for combining feature selection outcomes through a stochastic process," in *Knowledge Discovery in Databases: PKDD 2005*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2005, vol. 3721, pp. 667–674.

[34] C. Sumner, A. Byers, R. Boochever, and G. Park, "Predicting dark triad personality traits from Twitter usage and a linguistic analysis of tweets," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 2, 2012, pp. 386–393.

[35] Y. Takhteyev, A. Gruzd, and B. Wellman, "Geography of Twitter networks," *Social Networks*, vol. 34, no. 1, pp. 73–81, 2012, capturing Context: Integrating Spatial and Social Network Analyses. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0378873311000359

[36] Y. R. Tausczik and J. W. Pennebaker, "The psychological meaning of words: Liwc and computerized text analysis methods," *Journal of Language and Social Psychology*, vol. 29, no. 1, pp. 24–54, 2010. [Online]. Available: http://jls.sagepub.com/content/29/1/24.abstract

[37] K. Thomas, C. Grier, J. Ma, V. Paxson, and D. Song, "Design and evaluation of a real-time url spam filtering service," in *IEEE Symposium on Security and Privacy (SP)*. IEEE, 2011, pp. 447–462.

[38] K. Thomas, C. Grier, D. Song, and V. Paxson, "Suspended accounts in retrospect: an analysis of twitter spam," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*, ser. IMC '11. New York, NY, USA: ACM, 2011, pp. 243–258. [Online]. Available: http://doi.acm.org/10.1145/2068816.2068840

[39] H. Thomases, *Twitter Marketing: An Hour a Day*, ser. Serious skills. Wiley, 2009. [Online]. Available: http://books.google.com/books?id=oGBN0pyfkF4C

[40] A. M. Turing, "Computing machinery and intelligence," *Mind*, vol. 59, no. 236, pp. 433–460, 1950.

[41] V. G. Tusher, R. Tibshirani, and G. Chu, "Significance analysis of microarrays applied to the ionizing radiation response," *Proceedings of the National Academy of Sciences*, vol. 98, no. 9, pp. 5116–5121, 2001. [Online]. Available: http://www.pnas.org/content/98/9/5116.abstract

[42] J. Van Hulse, T. M. Khoshgoftaar, and A. Napolitano, "A comparative evaluation of feature ranking methods for high dimensional bioinformatics data," in *Information Reuse and Integration (IRI), 2011 IEEE International Conference on*, August 2011, pp. 315–320.

[43] J. Van Hulse, T. M. Khoshgoftaar, A. Napolitano, and R. Wald, "Feature selection with high-dimensional imbalanced data," in *IEEE International Conference on Data Mining Workshops, 2009. ICDMW '09.*, Y. Saygin, J. X. Yu, H. Kargupta, W. Wang, S. Ranka, P. S. Yu, and X. Wu, Eds. IEEE Computer Society, December 2009, pp. 507–514.

[44] C. Wagner, S. Mitter, C. Körner, and M. Strohmaier, "When social bots attack: Modeling susceptibility of users in online social networks," *Making Sense of Microposts (# MSM2012)*, p. 2, 2012.

[45] R. Wald, T. M. Khoshgoftaar, A. Napolitano, and C. Sumner, "Using Twitter content to predict psychopathy," in *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, vol. 2, 2012, pp. 394–401.

[46] R. Wallace, "The elements of AIML style," *Alice AI Foundation*, 2003.

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

[48] M. Wasikowski and X. wen Chen, "Combating the small sample class imbalance problem using feature selection," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1388–1400, October 2010.

[49] J. Weizenbaum, "Eliza — a computer program for the study of natural language communication between man and machine," *Commun. ACM*, vol. 9, no. 1, pp. 36–45, Jan. 1966. [Online]. Available: http://doi.acm.org/10.1145/365153.365168

[50] I. H. Witten and E. Frank, *Data Mining: Practical machine learning tools and techniques*, 2nd ed. Morgan Kaufmann, 2005.