Detection of Novel Social Bots by Ensembles of Specialized Classifiers

Mohsen Sayyadiharikandeh, Onur Varol, Kai-Cheng Yang, Alessandro Flammini and Filippo Menczer

1 Observatory on Social Media, Indiana University, Bloomington, IN, USA
2 Center for Complex Network Research, Northeastern University, MA, USA

ABSTRACT
Malicious actors create inauthentic social media accounts controlled in part by algorithms, known as social bots, to disseminate misinformation and agitate online discussion. While researchers have developed sophisticated methods to detect abuse, novel bots with diverse behaviors evade detection. We show that different types of bots are characterized by different behavioral features. As a result, commonly used supervised learning techniques suffer severe performance deterioration when attempting to detect behaviors not observed in the training data. Moreover, tuning these models to recognize novel bots requires retraining with a significant amount of new annotations, which are expensive to obtain. To address these issues, we propose a new supervised learning method that trains classifiers specialized for each class of bots and combines their decisions through the maximum rule. The ensemble of specialized classifiers (ESC) can better generalize, leading to an average improvement of 56% in F1 score for unseen accounts across datasets. Furthermore, novel bot behaviors are learned with fewer labeled examples during retraining. We are deploying ESC in the newest version of Botometer, a popular tool to detect social bots in the wild.

KEYWORDS
social media, bot detection, supervised learning

1 INTRODUCTION
Social media accounts partially controlled by algorithms, known as social bots, have been extensively studied [21, 41]. The automated nature of bots makes it easy to achieve scalability when spreading misinformation [36, 37], amplifying popularity [9, 34, 45], or polarizing online discussion [39]. Bot activity has been reported in different domains, including politics [5, 20, 39], health [2, 3, 6, 17], and business [12, 13]. Due to the wide adoption of social media, every aspect of people’s life from news consumption to presidential elections is vulnerable to potential manipulation by bots.

The public is beginning to recognize the existence and role of social bots: according to a recent Pew survey [40], two thirds of Americans have knowledge of bots and over 80% believe bots have a negative impact. Actions have been taken to restrict the potential damage caused by deceptive bots, as those that poses as humans. For example, California passed a “Bot Disclosure” law in July 2019, requiring bots to reveal themselves in certain cases (leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=201720180SB1001). However, there is no guarantee that such legislative solutions will be effective against malicious bots, or even that they will survive constitutional challenges. In an arms race between abusive behaviors and countermeasures, novel social bots emerge everyday and evade purge from the platforms [15, 21]. Therefore, the availability of tools to identify social bots is still important for protecting the authenticity and health of the information ecosystem.

Many social bot detection methods based on machine learning have been proposed in the past several years (see Related Work). Here we focus on supervised learning methods, particularly Botometer [43, 48], a widely adopted tool designed to evaluate Twitter accounts. Supervised methods are only as good as their training data. Bots with unseen characteristics are easily missed, as demonstrated by a drastic drop in recall when classifiers are faced with cross-domain accounts [18]. One common approach to address the lack of general model is to retrain models with new labeled datasets [48]. Unfortunately, high-quality datasets of annotated social media accounts are expensive to acquire.

In this paper, we aim to improve the cross domain performance of Botometer in the wild and better tune the method to the adversarial bot detection problem. Using classes of bot and human accounts in different datasets as distinct domains, we show that bot accounts exhibit greater heterogeneity in their discriminative behavioral features across domains, compared to human accounts. Motivated by this observation, we propose a novel method to construct a bot detection system capable of better generalization by training multiple classifiers specialized for different types of bots. Once these domain-specific classifiers are trained, unseen accounts are evaluated by combining their assessments. We evaluate cross-domain performance by testing on datasets that are not used for training, as opposed to in-domain evaluation through cross-validation. Without loss of in-domain accuracy, the proposed approach effectively increases the recall of cross-domain bots. It can also learn more efficiently from examples in new domains.

Given these results, the proposed method is being deployed in the newest version of Botometer, a widely adopted tool to detect social bots in the wild that is publicly available from the Observatory on Social Media at Indiana University.

2 THE CHALLENGE OF GENERALIZATION
2.1 Datasets
We considered various labeled datasets available through the Bot Repository (botometer.iuni.iu.edu/bot-repository). Most of the datasets are annotated by humans, while others are created using automated techniques based on account behavior, filters on metadata,
or more sophisticated procedures to achieve high precision (see Table 1 for details). For example, the vendor-purchased dataset contains fake followers purchased from several companies; accounts in cresci-stock show highly coordinated behavior that is extremely unlikely for human accounts; verified consists of randomly sampled verified accounts from the Twitter stream and is added to balance the datasets with only bots for our analysis. In addition to those datasets in the Bot Repository, we also collected accounts provided in a recent study conducted by Kaiser et al. [35]. These datasets made an assumption that all accounts belong to American and German politicians are being human accounts. They compliment this dataset with manually annotated German language bots and accounts listed in botwiki dataset.

For training models, we extract over 1,200 features in six categories: metadata from the accounts and friends, retweet/mention networks, temporal features, content information, as well as sentiment. These features are shown to be effective in identifying social bots and are described in detail in the literature [42, 43, 48].

### 2.2 Cross-domain performance comparison

Supervised bot detection methods achieve high accuracy based on in-domain cross-validation [4]. To measure how recall deteriorates in cross-domain evaluation, we perform an experiment using four datasets selected from Table 1: we train a model on one dataset and test it on another. We use Random Forest classifiers with 100 decision trees. The fraction of tweets outputting a positive label is calibrated using Platt’s scaling [31] and binarized with a threshold of 0.5. We use 5-fold cross-validation for in-domain classification; for consistency we split training and test samples in cross-domain cases as well, reporting average precision and recall.

The results of our experiment are shown in Fig. 1. Diagonal cells represent in-domain performance, off-diagonal cells report cross-domain performance. Both precision and recall tend to be higher for in-domain than cross-domain cases. This demonstrates limited generalization of supervised models across domains. The one exception is the high precision when testing on the cresci-17 domain, irrespective of the training datasets. This is due to the fact that cresci-17 includes spambots, which are represented in all datasets. By comparing the two panels, we see that recall of bots is more impaired in cross-domain tests, which aligns with previous findings [11, 18]. The method proposed in this paper improves cross-domain bot recall.

To interpret the cross-domain classification results, we plot the distributions of bot scores in Fig. 2. A bot score is the output of a Random Forest classifier and corresponds to the proportion of decision trees in the ensemble that categorize the account as a bot. In the diagonal plots (in-domain tests), the density plots are left-skewed for humans and right-skewed for bots, representing a good separation and yielding high precision, recall, and F1. For most of the cross-domain experiments, the score distributions are still left-skewed for humans, but not right-skewed for bots. This suggests that bot accounts tend to have lower scores in cross-domain cases, resulting in lower recall. Human accounts, on the other hand, exhibit consistent characteristics across datasets. We will rely on this observation to improve the generalization power of the proposed method.

### 2.3 Predictability of different bot classes

There are different kinds of bots. To illustrate, consider the three different bot classes in the cresci-17 dataset: traditional spambots, social spambots, and fake followers. We trained decision tree models to discriminate each of these classes of bots from the others. Table 2 lists the most informative features of each class revealed by the models. We observe different characteristics: traditional spambots generate a lot of content promoting products and can be detected by the frequent use of adjectives; social spambots tend to attack or support political candidates, therefore sentiment is an informative signal; finally, fake followers tend to have aggressive following patterns, flagged by the friend/follower ratio.

Given such heterogeneity of bot behaviors, we conjecture that the drop in cross-domain recall can be attributed to the different discriminating features of accounts in different datasets. Let us explore this conjecture using caverlee and varol-icwsm as two
Figure 2: Bot score distributions for human (blue) and bot (red) accounts in each experiment. Models used in these experiments are trained on the datasets labeled along the rows and tested on the datasets listed in the columns.

Table 2: Most informative features per bot class in cresci-17.

| Rank | Traditional spambots | Social spambots | Fake followers |
|------|----------------------|-----------------|---------------|
| 1    | Std. deviation of adjective frequency | Tweet sentiment arousal entropy | Max. friend-follower ratio |
| 2    | Mean follower count | Mean friend count | Std. deviation of tweet inter-event time |
| 3    | Tweet content word entropy | Mean adjective frequency | Mean follower count |
| 4    | Max. friend-follower ratio | Minimum favorite count | User tweet-retweet ratio |
| 5    | Max. number of retweet count | Tweet content word entropy | Mean tweet sentiment happiness |

Figure 3: Separation of bots and humans based on two features in different datasets. Both plots show the logistic-regression decision boundary obtained from caverlee. Exemplar datasets. We train a Random Forest in the binary classification mode on caverlee and use the Gini impurity score of the decision trees to find the two most informative features. Fig. 3 visualizes bot and human accounts in caverlee and varol-icwsm on the plane defined by those two features. For the in-domain case (left scatter plot in the figure), the linear classifier is sufficient to separate human and bot accounts. But in the cross-domain case (right plot), the same linear model fails, explaining the drop in recall: different features and distinct decision rules are needed to detect different classes of bots.

3 METHODS

3.1 Baseline bot detection models

Before presenting the proposed method, let us select two baselines for evaluation. The first baseline is the current version of Botometer, often considered the state-of-the-art method for bot detection [48].
The model is a Random Forest, a classification model that has proven to be effective in high-dimensional problems with complex decision boundaries. In this approach, we output the fraction of positive votes as a bot score. Bots from all datasets are merged into a single bot (positive) class. We refer to this baseline as Botometer-v3. We also consider a variation of the Botometer baseline that does not consider a set of features describing time zone and language metadata, as those are no longer available through the Twitter API. We refer to this variation as Botometer-v3.1. The two Botometer baselines use 1,209 and 1,160 features, respectively.

We use tweetbotornot2 as a second baseline. This model is based on a supervised classifier that considers over a hundred features in three main categories: user-level attributes, tweeting statistics, and text-based patterns. The motivation behind this choice of baseline is that tweetbotornot2 has been developed independently, is widely used, and is easily accessible by the general public via an R library (tweetbotornot2.mikewk.com).

### 3.2 Proposed method

The proposed approach is inspired by the two empirical findings discussed in the previous section. First, inspired by the observation that human accounts are more homogeneous than bots across domains, we train a model on all human and bot examples across datasets and use it to identify likely humans. Second, since different bot classes have different sets of informative features, we propose to build specialized models for distinct bot classes. The specialized human and bot models are aggregated into an ensemble and their outputs are combined through a voting scheme. We call this approach Ensemble of Specialized Classifiers (ESC) and we also refer it as the Botometer-v4.

Fig. 4 illustrates the ESC architecture. The human detection subsystem actually corresponds to Botometer (the baseline classifier), and constitutes the left-most component $RF_0$ of the ensemble shown in the figure. We then build specialized bot classifiers using Random Forest models $(RF_1, \ldots, RF_n$ in Fig. 4). All Random Forest classifiers in this paper are implemented using the scikit-learn package [33]. We use 100 decision tree estimators; all other parameters take the default values. Each specialized classifier $RF_i$ is trained on a balanced set of accounts from bot class $BC_i$ and an equal number $|BC_i|$ of human examples sampled from human accounts across all datasets.

![Figure 4: Illustration of the proposed model. The bot score from $RF_0$ corresponds to the previous version of Botometer, that from the voting module to the new (ESC) version.](Image)

A bot score is calculated by a voting scheme for the classifiers in the ensemble. Among the specialized bot classifiers, the one that outputs the highest bot score $s_i$ is most likely to have recognized a bot of the corresponding class. Therefore we use the maximum rule to aggregate the bot scores. For the human classifier $RF_0$, a low bot score $s_0$ is a strong signal of a human account. Therefore we determine the winning class as $i^* = \arg \max_i \{s_i^*\}$ where

$$s_i^* = \begin{cases} 1 - s_i & \text{if } i = 0 \\ s_i & \text{else.} \end{cases}$$

The bot score returned by ESC is obtained by calibrating the score $s_i^*$ using Platt’s scaling [31]. As we see in the Results section, the maximum rule has the effect of shifting scores of likely bots toward one and the scores of likely humans toward zero. Along with the bot score, ESC can also produce the bot class label $i^*$ as an explanatory outcome.

### 4 RESULTS

#### 4.1 In-domain performance

Before discussing cross-domain performance, let us demonstrate that ESC is capable of detecting bots with good accuracy in the classic (in-domain) scenario. Using 5-fold cross-validation, ESC achieves an Area Under the ROC Curve (AUC) of 0.96, similar to the Botometer classifier (0.97 AUC). The scatter plot in Fig. 5 shows a good agreement between the bot scores obtained with ESC ($Botometer-v4$) and the Botometer-v3 baseline (Spearman’s $\rho = 0.87$).

Let us pay special attention to accounts having low Botometer and high ESC scores, in other words those that are likely human according to Botometer, but likely bots according to ESC (region highlighted in Fig 5). These are the only cases where we observe
a clear disagreement between the two methods on ground-truth labels. We focus on 332 accounts in this region that are labeled as human (75% are from caverlee), which represent approximately 1% of the examples labeled as human in the training data. One possible interpretation of the disagreement is that these accounts are incorrectly classified by ESC (false positives). Another possibility is that some of these accounts may have changed since they were manually labeled. Indeed, training datasets are subject to change over time as accounts become inactive, suspended, or get compromised by third-party applications. Such changes could lead to errors on ground-truth labels.

Manual inspection of a random sample of 50 of the accounts highlighted in Fig 5 reveals that the human labels are no longer accurate for most of them — they have been inactive for years, are currently devoted to spam diffusion, and/or are controlled by third-party applications. This suggests that ESC can identify impurities in the training data. While mislabeled accounts impair the performance of machine learning models, we conjecture that the ESC model is still able to recognize them because it is more robust to errors — the incorrect labels only affect a subset of the classifiers.

4.2 Cross-domain performance

We want to demonstrate that the proposed ESC approach generalizes better to cross-domain accounts compared to the current version of Botometer.

To train the specialized classifiers, we need coherent bot behaviors. To this end, we organized the bot accounts in the training data into separate classes of bots: simple bots, spammer bots, fake-followers, honest bots, political bots, and other bots. Bot accounts in the caverlee dataset are assigned to simple bots. For spammer bots we use bot accounts in pron-bots and cresci-17. Fake-followers include subset of bot accounts in cresci-17 and vendor-purchased. The honest and political bot categories are derived from the botwiki and political-bots dataset, respectively. The rest of the bots captured in other datasets are aggregated into the other bots category.

In this set of experiments, some datasets are held out in the training phase and are then used as cross-domain test cases: cresci-stock, gilani-17, cresci-rtbust, kaiser-1, kaiser-2, and kaiser-3. In addition, combined-test combines these datasets while avoiding duplication. We focus on F1 and AUC as the accuracy metrics, using 5-fold cross-validation to provide confidence intervals.

We compare the ESC approach (Botometer-v4) with the Botometer-v3 baseline model [48], as well as the Botometer-v3.1 baseline to exclude the possibility that improvements are due to the removal of features based on deprecated metadata. The language agnostics version of Botometer (excluding English-based linguistic features) is used on the kaiser-2 dataset because of its German tweet content.

To illustrate the main enhancement afforded by ESC, let us compare the distributions of bot scores generated by Botometer-v3 and Botometer-v4 in cross-domain experiments. Due to space limitations, we illustrate in Fig. 6 just one case where cresci-rtbust is used as test set; other cases are similar. Both methods tend to yield low scores for human accounts, as the same classifier (RFv3) is used. On the other hand, Botometer-v4 produces significantly higher scores than Botometer-v3 on bot accounts. This is a result of the maximum rule and leads to higher cross-domain recall, or better generalization.

Fig. 7 shows that Botometer-v4 (ESC) outperforms the Botometer baseline in most cases. On average across the six datasets, recall goes from 42% to 84% (an improvement of 100%) while precision increases from 52% to 64%. As a result, F1 increases from 47% to 73% (an improvement of 56%). On the combined-test dataset, recall goes from 77% to 86%, precision stays at 70%, and F1 goes from 73% to 77% (an improvement of 5%). Comparisons based on AUC scores are similar.

The kaiser-1 and kaiser-3 datasets include bots from botwiki, which are part of the Botometer-v4 training data. Therefore these two cannot be considered completely hold-out datasets, but are
included nonetheless because they were used to highlight weaknesses of Botometer-v3 in a recent independent report [35], so they provide us with an opportunity to demonstrate the performance of the latest Botometer model. Even if we exclude botwiki1 from the training data, ESC still outperforms Botometer-v3. For example, it yields an F1 score of 0.84 on kaiser-1 and 0.80 on kaiser-3.

Botometer-v4 yields F1 better than or comparable with tweetbotornot2 on all datasets except those from Rauchfleisch and Kaiser [35]. The AUC metric is comparable on those datasets, while tweetbotornot2 wins on gillani-17 and Botometer-v4 wins on the other datasets. On the combined-test dataset, Botometer-v4 outperforms tweetbotornot2 on both F1 and AUC.

In interpreting these results, note that kaiser-3 contains human accounts from kaiser-2 and bot accounts from kaiser-1, so they are not independent. kaiser-2 includes only 27 bot accounts; the F1 score is sensitive to this class imbalance. Furthermore, while we do not know how the tweetbotornot2 baseline was trained, it uses a feature for “verified” profiles and tends to assign a low bot score to them, even automated ones such as @twittersupport. 98% and 72% of accounts labeled human in kaiser-1 and kaiser-3 respectively are verified. This biases the results in favor of tweetbotornot2.

4.3 Model adaptation

Real-world applications of social bot detection always face the challenge of recognizing novel behaviors that are not represented in the training data. Periodic retraining to include newly annotated datasets helps systems adapt to these unseen examples. A common approach is to train a new classifier from scratch including both old and new training data, which may not be efficient. The proposed ESC method alleviates this problem because we can add a new classifier $RF_{n+1}$ to the ensemble to be trained with the new data, without retraining the existing classifiers.

Let analyze how quickly the Botometer-v3 and ESC models adapt to a new domain. To quantify this, we split the data from a hold-out domain into two random subsets for training and testing. Results are presented using varol-icwsm as the hold-out domain for both models. (We reach similar conclusions using other hold-out datasets.) 800 examples from the hold-out dataset are used for training. In each iteration we randomly sample 50 examples and add them to the training set. In the Botometer-v3 case, we retrain the entire classifier ($RF_5$), whereas in the ESC case we only train a newly added specialized classifier ($RF_{n+1}$).

Fig. 8 shows how the F1 score on the test data improves as a function of the number of training examples from the hold-out domain. The Botometer-v3 baseline adapts more slowly. We can interpret this result by recalling that the size of the hold-out dataset is small compared to the training size (over 67,000 examples in Botometer-v3). The decision trees use Gini gain as a feature selection criterion, therefore the number of examples sharing the same informative features affects the selection of those features. As a result, the old bot classes with more examples dominate and the classifier struggles to learn about the new domain. On the other hand, the ESC architecture quickly learns about new bots through the new classifier, which starts from scratch, while what was learned about the old ones is preserved in the existing classifiers. This means that fewer labeled examples are needed to train a new specialized classifier when novel types of bots are observed in the wild.

5 RELATED WORK

Different approaches have been proposed for automatic social bot detection. Crowdsourcing is convenient and one of the first proposals to collect annotated data effectively [47], but annotation has limitations due to scalability and user privacy. Thus automatic methods are of greater interest, especially for social media services that deal with millions of accounts. The structure of the social graph captures valuable connectivity information. Facebook employed an immune system [38] that relied on the assumption that sybil accounts tend to connect mostly to other sybil accounts while having a small number of links to legitimate accounts.

Supervised machine learning approaches, such as the one proposed in this paper, extract various features from an account’s profile, social network, and content [16, 23, 29, 41, 43, 48, 49]. These methods rely on annotated datasets for learning the difference between bot and human accounts. However, since bots change and evolve continuously to evade detection, supervised learning algorithms need to adapt to new classes of bots [21, 44, 48].

Some unsupervised learning methods have been proposed in the literature [11, 27]. They can be less vulnerable to performance decay across domains. They are especially suitable for finding coordination among bots [1, 29, 32]. Since accounts in a coordinated botnet may not appear suspicious when considered individually, supervised methods would miss them [11, 24]. Identifying botnets requires analysis of the activity of multiple accounts to reveal their coordination. Depending on the type of bots, similarity can be detected through tweet content [8, 25], temporal features in the timelines [7, 10], or retweeting behavior [28, 46].

A recent research direction is to address the limits of current bot detection frameworks in an adversarial setting. Cresci et al. [14] predict that future techniques will be able to anticipate the ever-evolving spambots rather than taking countermeasures only after seeing them. The performance decay of current detection systems in the wild was reported by Cresci et al. [11], who showed that Twitter,
human annotators, and state-of-the-art bot detection systems failed at discriminating between some new social spambots and genuine accounts. In agreement with the present findings, Echeverría and Zhou [19] suggest that detecting different types of bots requires different types of features. Echeverría et al. [18] proposed the leave-one-botnet-out methodology, to highlight how detection methods do not generalize well to unseen bot classes due to bias in the training data. Even a classifier trained on 1.5 million accounts and 22 classes of bots is incapable of identifying new classes of bots. Here we follow the idea of leave-one-class-out methodology by leaving the complete test cases out of the training sets to evaluate the generalization power of the proposed approach.

Some papers have characterized bot classes [30]. Lee et al. [26] define traditional spammers, mention spammers, friend infiltrators, and social spammers in their dataset. Cresci et al. [11] highlight that it is hard for human annotators to assign one label to one account to describe its bot class. They also report that some bot classes, like social spambots, are hard to distinguish from human accounts. Despite the existence of different types of bots, no systems have been presented in the literature to automatically identify the type of a bot as the method proposed here.

Rauchfleisch and Kaiser [35] have criticized Botometer (more specifically Botometer-v3) for its high false positive and false negative rates. As we have discussed in this paper, we share these concerns and acknowledge that like any supervised learning model, Botometer makes mistakes. Indeed, the new version introduced in the paper is motivated by this issue and partly addresses it by improving cross-domain recall (false negatives). At the same time, Rauchfleisch and Kaiser [35] may overestimate the false positive rate by assuming that no politician account uses automation. We believe this assumption is not realistic, considering these accounts are often managed by media teams and use scheduling tools for content creation. In fact, Botometer currently does not use the “verified” status as a feature because it could lead to false negatives [45]. Another source of bias is that Rauchfleisch and Kaiser [35] overlook accounts that are no longer available due to suspension, possibly leading to an underestimation of both precision and recall.

6 CONCLUSION
The dynamic nature of social media creates challenges for machine learning systems making inferences and predictions on online data. On the one hand, platforms can change features, require models to be retrained. Further difficulties arise as accounts used for training change behavior, become inactive, compromised, or are removed from the platform, invalidating ground-truth data. On the other hand, account behaviors can change and evolve. As is typical in adversarial settings, automated accounts become more sophisticated to evade detection. The emergence of more advanced bot capabilities brings additional challenges for existing systems that struggle to generalize to novel behaviors.

Despite impressive results when training and test sets are from the same domain — even using cross-validation — supervised models will miss new classes of bots, leading to low recall. We demonstrate in this paper that the performance deterioration observed for out-of-domain accounts is due to heterogeneous bot behaviors that require different informative subsets of features. Inspired by this, we presented a novel approach for the automatic identification of novel social bots through an ensemble of specialized classifiers trained on different bot classes. We demonstrated empirically that our proposed approach generalizes better than a monolithic classifier and is more robust to mislabeled training examples.

The proposed architecture is highly modular as each specialized classifier works independently, so one can substitute any part with different models as needed. We can also include additional specialized classifiers when new annotated datasets become available. We showed that this approach allows the system to learn about new domains in an efficient way, in the sense that fewer annotated examples are necessary. In addition, the architecture of our model offers greater transparency about the decision process by allowing inspection of the outputs of different ensemble components and by providing interpretable class labels in output.

In light of these results, we are deploying ESC in the newest version of Botometer, a tool to detect social bots in the wild that is available through the Observatory on Social Media at Indiana University. Botometer can be accessed both through an interactive website (botometer.org) and programmatically through a public API (rapidapi.com/OSoMe/api/botometer-pro). The tool is quite popular among researchers and practitioners; at the time of this writing, it is mentioned in 1,290 articles according to Google Scholar, and it fields over half a million queries daily (Fig. 9).

In future work, we would like to investigate methods to recognize the appearance of a new type of bots that warrants the addition of a new classifier to the ensemble. We could also design an unsupervised method to cluster similar accounts across datasets automatically, and assign homogeneous accounts to each of the specialized classifiers. Finally, we can design active learning query strategies to make the retraining process even more efficient than with random selection. This would be useful when reliable user feedback is available to be used as an oracle.

ACKNOWLEDGMENTS
This work was supported in part by DARPA (grant W911NF-17-C-0094), Knight Foundation, and Craig Newmark Philanthropies. We are grateful to Clayton A. Davis and Emilio Ferrara who contributed to early versions of Botometer.
