Social Network Analysis and Mining (2024) 14:220
https://doi.org/10.1007/s13278-024-01387-7

ORIGINAL ARTICLE

Adversarial botometer: adversarial analysis for social bot detection

Shaghayegh Najari[1] · Davood Rafiei[2] · Mostafa Salehi[1,3] · Reza Farahbakhsh[4]

Received: 25 July 2024 / Revised: 25 October 2024 / Accepted: 16 November 2024

Abstract
Social bots have become increasingly prominent in Online Social Networks, imitating human behavior and raising concerns about their deceptive capabilities. With advances in Generative AI, these bots are now able to generate highly realistic and complex content, making detection a significant challenge. While various bot detection approaches exist, their effectiveness has not been thoroughly evaluated. In this study, we examine the behavior of a text-based bot detector across three key scenarios: adversarial interactions between bots and detectors, attack examples generated by bots to poison datasets, and cross-domain analysis of different types of bots. Our findings show that detection performance varies significantly across bot types. Models trained on commercial bots struggle to detect attack samples accurately, while models trained on political and financial bots perform better. Furthermore, weak discriminators in the detection model can lead to issues like mode collapse, which can be addressed by employing techniques such as autoencoders, Energy-based GANs, or stronger cost functions. This analysis highlights the need to better understand feature importance in bot detection models, as well as to refine pre-processing steps to improve detection accuracy across different social bot domains.

Keywords Social bot detection · Adversarial training · Conversational models

1 Introduction

Along with the development of Artificial Intelligence (AI) and since it enters to bots like Joseph Weizenbaum's ELIZA for emulating a Rogerian psychotherapist, many things have changed

The utilization of these automated algorithms in social networks, commonly known as social bots may initially have some positive effects; however, as time passes, their activities can become increasingly destructive.

In 2017, the average presence of bots on active Twitter accounts was estimated to be around 15% (Varol et al. 2017), while on Facebook, it was approximately 11% in 2019 (Zago et al. 2019). These numbers indicate a considerable share of automated accounts on both platforms. Moreover, the presence of bots tends to increase significantly when there are strong political or economic interests involved. A study conducted in 2019 revealed that 71% of Twitter users discussing trending US stocks were likely to be bots (Cresci et al. 2019). Similar results were found regarding the presence of bots in online cryptocurrency discussions (Nizzoli et al. 2020) and their involvement in spreading "infodemics" during the COVID-19 pandemic (Gallotti et al. 2020).

In recent years, the rapid development of new models in GenAI leads to the emergence of powerful transformer-based bots such as Generative Pre-trained Transformer (GPT) (Vaswani et al. 2017). These advancements have enabled social bots to engage in more complex interactions and penetrate popular discussions, such as participating in

✉ Mostafa Salehi
Mostafa_salehi@ut.ac.ir

Shaghayegh Najari
najari.shaghayegh@ut.ac.ir

Davood Rafiei
drafiei@ualberta.ca

Reza Farahbakhsh
reza.farahbakhsh@it-sudparis.eu

[1] Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

[2] Computing Science Department, University of Alberta, Edmonton, Alberta, Canada

[3] School of Computer Science, Institute for Research in Fundamental Science (IPM), P.O. Box 19395-5746, Tehran, Iran

[4] Institut Polytechnique de Paris, Telecom SudParis, Evry, France

Published online: 24 November 2024

entertaining conversations, leaving comments on posts, and responding to questions (Ferrara et al. 2016).

In response to detrimental activities of social bots, extensive research has been devoted to identification and mitigation of social bots. However, a major challenge with bot detection is the poor performance of the models under more complex circumstances, especially when a social bot employs deceptive tactics. This is mainly because a social bot detector is considered a fixed module without any progress, and thus the data on which the model is trained on is also considered fixed (De Nicola et al. 2021). This is problematic when the data that the model is trained on fails to capture unforeseen or future patterns. Even a small irregularity in the training data can cause the model's performance to drop significantly.

Evaluating the behavior of bot detection models in the presence of attack examples generated by human-like bots is an under-researched area. This study aims to address this gap by exploring a novel approach wherein a bot engages in an adversarial game with a bot detection model. In this game, where a bot automatically generates perturbations, the performance of bot detection model can be evaluated using a process called adversarial training (Goodfellow et al. 2014b). In particular, we formulate adversarial scenarios in which a bot simulates human behavior to generate attack examples. In contrast, a bot detection model is then tasked with distinguishing between real and fake examples. With this approach, we can effectively evaluate the behavior of bot detection models to detect and defend against attacks by fraudulent bots.

Our evaluation, conducted on two benchmark datasets: Midterm-2018 and Cresci-2017 that includes three different categories of social bots and one category of human users. Through analyzing the results, we have reached some achievements that could utilize for the future works.

The main contributions of this study can be summarized as follows:

- We model a social bot as an interactive and automated conversational model.
- To evaluate the bot detector's behavior in a dynamic condition, we design an adversarial game between bot detector and the bot that is producing some adversarial attacks.
- To thoroughly evaluate the performance of the bot detector, we ran 3 different scenarios and presented our achievements, which we can use for future work.

The rest of the paper is organized as follows. In Sect. 2, we review the literature related to our work. Section 3 presents our proposed model and scenarios and Sect. 4 presents our results and analysis. Finally, we conclude the work with a discussion of some future research directions in Sect. 5.

2 Related works

To evaluate the performance of the bot detection module when faced with a complex social bot that mimics human behavior, it is important to have an overview of generative bots and the bot detectors. Therefore, we will discuss them in the following sections.

2.1 Bot generation

As automated computer programs, bots have existed in various forms since the early days of computers. These forms range from those controlled by humans, such as spam generators (Wang 2010), to fully automated algorithms like chatbots. Advances in natural language processing, particularly the use of simple neural networks and transfer learning-based models (Weiss et al. 2016), have enabled bots to engage with real users and generate text that closely resembles human text. This has led to a major challenge in distinguishing between content generated by bots and content created by humans (Alarifi et al. 2016).

One type of automated generative models that is designed to behave like a human is a dialogue model (Xu et al. 2021). This is a model that is able to capture the structure and meaning of conversations between two entities, typically a user and an interactive computer system. Dialogue systems can be categorized into two main classes (Chen et al. 2017): task-oriented and non-task-oriented (also referred to as open-ended conversational agents). Task-oriented dialogue systems are developed to accomplish specific tasks, whereas non-task-oriented systems are more versatile and capable of engaging in broader and more general conversations. A widely adopted approach for constructing dialogue models is the utilization of a sequence-to-sequence model (Li et al. 2015). This model comprises an encoder component responsible for mapping the input sequence to an intermediate vector, and a decoder component that generates a response utilizing the hidden state of the encoder and the intermediate information obtained.

In the recent decade, Generative Adversarial Network (GAN) as a generative model designed for sequential data such as image generation (Goodfellow et al. 2014a), some later they were adapted to process discrete and textual data. SeqGAN is an example of a GAN, which has been specifically adapted for text generation (Yu et al. 2017). Here, both the generator and the discriminator work together to improve the quality of the text produced by the generator. The generator makes decisions at each time step to maximize expected rewards, which are determined by the discriminator. (Yu et al. 2017).

As the goal of this paper is to analyze the behavior of bot and bot detection in a tug-of-war, we utilized the dialogue models beside SeqGAN to design some competition scenarios based on.

2.2 Bot detection

In general, contributions to analyze the bot detectors behavior could to fall into two main categories: feature-based and model-based approaches.

In Feature-based category, social bots are identified by some simple and well-known machine learning models, such as Random Forest and Support Vector Machine, by analysing a set of behavioural features extracted from the input dataset (Heidari et al. 2021; Aljabri et al. 2023; Orabi et al. 2020), directly. Thus, the performance of these models are more dependent on the features selected.

Features can be divided into two main classes: content-based and user-based. Content-based features focus on the content of the behaviour-based post, including the text of a tweet or linguistic annotations such as part-of-speech tags (Jr et al. 2018). User-based features, on the other hand, are based on a user's profile, behavior (Velayutham and Tiwari 2017), and connections to other users in the network (Dorri et al. 2018).

In another category, models are improving to extract more informative features from the original contexts. Recently, some models such as deep learning-based techniques have emerged that attempt to improve social bot detectors by automatically extracting features and feeding them into a deep neural network that enables the extraction of more informative feature vectors. By using multiple layers of interconnected neurons, deep learning models can capture intricate patterns and representations in the data, leading to more effective feature extraction (Hayawi et al. 2023).

Long Short-Term Memory (LSTM) and Convolutional Neural Networks (CNN) are popular neural network architectures that have achieved successful results in tasks such as bot detection (Kudugunta and Ferrara 2018; Beskow and Carley 2019; Arin and Kutlu 2023). In this area, a novel deep neural network model named RGA (ResNet, GRU, and Attention) is developed to address the task of detecting social bots. The RGA model combines the power of a residual network (ResNet), a bidirectional gated recurrent unit (BiGRU), and an attention mechanism (Wu et al. 2021). These methods often represent the raw text as a vector and use pre-trained linguistic models such as Global Vectors for Word Representation (GloVE Pennington et al. 2014) to encode initial features.

2.3 Existing research GAP and objective

With the advance of bots through recent developments in genAI, there is a pressing need for powerful social bot detection methods in social networks to understand better the influence and impact of these bots and develop effective countermeasures (Ferrara 2023).The main difference between existing GenAI-based bots and bots trained with the proposed GAN-like framework is how they are designed to avoid detection and create realistic responses. Traditional GenAI-based bots aim to produce human-like content using large language models, but they are not specifically trained to avoid detection. These bots may create realistic responses, but they can still show patterns that detection models, especially well-trained ones, can recognize. On the other hand, bots trained with the proposed GAN-like framework use adversarial learning. In this setup, the bot (generator) improves by making its responses more deceptive to trick the detector (discriminator). Meanwhile, the detector gets better at spotting bot behavior through continuous interactions with the bot. This back-and-forth process helps both improve. The GAN-like framework focuses not only on creating realistic content but also on helping bots avoid detection, making it harder for traditional detectors to catch them.

As previous studies have shown, there is a lack of effective studies to evaluate social bot detection methods when confronted with intelligent bots that attempt to deceive the detectors. This was a strong incentive for us to take a new perspective and try to address this problem in more detail.

Most of the previous studies use the GAN framework to augment the lower class data such as bot (Najari et al. 2022), fake reviews (Aghakhani et al. 2018), or spam (Shehnepoor et al. 2021) to make the detectors stronger. Here, we would like to use GAN not as an auxiliary module but as an adversary module to design a game between bot and bot detector to analyze their behavior and strength under dynamic conditions.

Specifically, we use GAN to extend the bot class not by bot-like samples, but to include examples generated by human or human-like models: Human-like examples are generated by the adversarial bot (pre-trained on the class bot and fine-tuned on the class human), which knows the bot detection results as feedback. This extension will allow the evaluation of bot detection models in terms of their ability to distinguish between bots and humans in more complex states. In this study, we focus on evaluating the effectiveness of bot detection models in the presence of attack examples produced by generative model.

3 Proposed method: adversarial botometer

In this part, we have review the proposed scenarios.

3.1 Scenario 1: adversarial game between bot and bot detection

Figure 1 presents an overview of our proposed framework, which entails the design of an adversarial game between a bot and bot detection model. In this setup implemented by GAN, the bot is responsible for generating samples that resemble human behavior. On the other hand, the bot detection model tries to distinguish between the patterns generated by the bot and those generated by humans. Through this adversarial interaction, the generative model seeks to improve its ability to generate samples that are indistinguishable from those generated by humans, while the discriminator aims to accurately identify bot-generated samples.

To design this synthetic game, it is first necessary to specify a generative model that plays the role of the bot, and a discriminator model that plays the role of the bot detector. As shown in the previous section, there are a wide range of models to do this selection. Here, we select the Seq2Seq model as the generative model and the Contextual-LSTM as the discriminator model for several reasons:

- *Bot Generation: Seq2Seq Model* With the growth of bots through recent developments in GenAI, the recent social bots are not rigid generative models; rather, they are able to interact with other users, mimic their behaviour. To address this, we chose the seq2seq model as the basis for dialogue models.
- *Bot Detection: Contextual-LSTM* Since in our problem a bot tries to generate text like a human, the generated content is text. As the text is sequential data and LSTM is known to be the best example of DL-based models for sequential data, we chose Contextual-LSTM as the basis for text-based bot detectors.

The training phase for the bot and bot detection models consists of two main steps: Pre-training, and Adversarial-training.

3.1.1 Pre-training phase

In the context of dialog settings, our main focus is on generating answers to questions or comments based on observed posts. To achieve this, we use a sequence-to-sequence model consisting of an encoder and a decoder.

The encoder component captures an encoded representation of the input question or comment to help the model to understand the input and extract relevant features. The decoder component takes the encoded representation from the encoder and generates a response, in return.

Formally speaking, given a corpus of message pairs (S, R) where the source message S, consists of tokens $s_1, ..., s_n$ and the response (target) message R, consists of tokens $r_1, ..., r_m$, the bot is trained to maximize the total log likelihood of observed target messages, given their respective source messages:

$$\sum_{(S,R)} \log P(r_1, ..., r_m | s_1, ..., s_n) \tag{1}$$

Here, the goal of decoder D is to find an approximated distribution of trainset conditioned on generator parameter θ, and initial source S. Then, the decoder produces each token of samples based on preceding ones as shown in eq. 2.

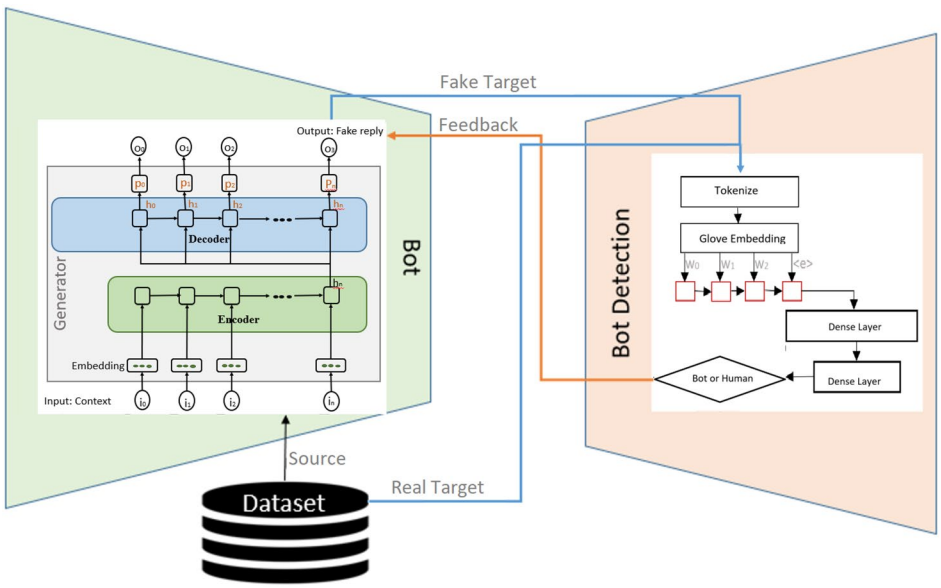


Fig. 1 Proposed adversarial game

$$D_\theta(R_{1:T}|\theta, S) = \prod_{t:1,2,...,T} D_\theta(r_t|r_{1:t-1}, \theta, S) \quad (2)$$

As illustrated in Fig. 1, the bot detector takes two types of inputs: fake targets generated by the generative model, or real ones selected from the target dataset. Given a corpus of pair message (S, R), score $y = 1$ if R was sampled from the training data and $y = 0$ otherwise, this model is trained to maximize the probability of correct labeling as follows:

$$\sum_{(y,S,R)} \log P(y|s_1, ..., s_m, r_1, ..., r_n) \quad (3)$$

3.1.2 Adversarial training

During the adversarial training phase, the goal is to generate samples that have higher realism, resulting in higher rewards. Here, a mismatch arises between the generative bot that produces sentences token by token and the discriminator model that evaluates the entire generated sequence.

To solve this problem, several approaches have been proposed previously. One effective method is to use a Monte Carlo Search Tree (MCST). MCST uses a tree-based search algorithm that explores the potential upcoming sequences (Yu et al. 2017). The signaling feedbacks obtained from the discriminator can then be propagated backwards to the generator. The minimax objective function of this adversarial game could be as follows:

$$min_B\_max_{BD}(E_{(s,r)\sim P_{(S,R)}} \log BD(r) + \log(1 - BD(B(r|s)))) \quad (4)$$

Here, B and BD demonstrate the bot and bot detector models, respectively. B generates samples satisfying $P_B(r|s)$; where, the generator B as the bot generate sample r selected from the response set R and given s selected from the source set S. Practically, we use a Long Short Term Memory (LSTM) to generate words, where each recurrent unit has embedding size 25, hidden dimension and a batch size of 64.

In this game, the bot detector (BD) provide a feedback to the bot (B). This feedback serves as an indicator of the bot's success or failure in fooling the bot detector. Based on this information, the bot regulates its generation strategy and produces more patterns, which are then sent to the bot detector for evaluation. This iterative process continues in a loop, forming a dynamic game between the bot and the bot detection model. Algorithm 1 provides more details about the process of this game.

As shown in Algorithm 1, the proposed framework first normalizes and pre-processes inputs (Kudugunta and Ferrara 2018) in order to prepare tweets for input to the LSTM network. This pre-processing phase involves removing punctuation, tokenzing the tweets using the methods from Global Vectors for Word Representation (GloVE) (Pennington et al. 2014)

We evaluate this framework through both live and offline adversarial game, and the results are reported in the next section.

Algorithm 1 Adversarial game of bot and bot detection models.

Require: Bot detector BD, Bot B, Dataset X containing pairs of (s, t), as a symbol of (Source, Target)
Initialize B, BD parameters with the random weights
Pre-process and tokenize pairs (s, t) available in X
Pre-train B and BD
for Training Iterations **do**
  for BD-training-steps **do**
    Sample t from the dataset X
    Sample $\hat{t} \sim B(s)$
    Update BD using (t) as positive sample and $(\hat{t})$ as negative sample using Eq. 3
  end for
  for B-training-steps **do**
    Sample (s,t) from the dataset X
    Sample $\hat{t} \sim B(s)$
    Update BD using (t) as positive sample and $(\hat{t})$ as negative sample and use output of BD as reward $BD(B(s))$ in Eq. 3.1.2
    Update B using defined objective function in Eq. 3.1.2
  end for
end for

3.2 Scenario 2: data poisoning

To assess the effectiveness of our bot detection model on a more complicated dataset that include examples of attacks, we adopt two distinct approaches. Firstly, we can select attack examples directly from the dataset itself. Alternatively, we can generate attack examples by simulating bot behavior using the GAN framework.

In the first approach, we examine the existing dataset and specifically identify instances that illustrate attack behaviors. These examples may involve various forms of malicious activities, such as spamming, misinformation propagation, or coordinated manipulation. By incorporating these attack examples into our evaluation, we can evaluate the robustness and accuracy of our bot detection model in detecting and classifying these malicious behaviors.

The second approach is to generate attack patterns by simulating the behavior of the bot. Using the GAN technique and methods described earlier, we can simulate the actions and patterns of bots involved in attacks. By generating synthetic attack patterns, we can create a controlled environment to comprehensively evaluate the performance of our bot detection model in identifying and distinguishing between normal user behavior and malicious bot behavior.

Both approaches provide valuable insights into the effectiveness of our bot detection model in processing complex datasets of attack patterns. By combining real-world attacks with simulated attack scenarios, we can thoroughly evaluate the model's capabilities and improve its ability to detect and mitigate various forms of bot-driven attacks.

3.3 Scenario3: domain and model explanation

To comprehensively evaluate the textual distinctions among social bots, we carried experiments on 14 NLP features extracted from each class of social bot datasets. These features included mention count, hashtag count, stopwords count, word count, unique word count, quoted word count, character count, sentence count, capital character count, capital word count, unique-to-total word ratio, average sentence length, average word length, and stopword-to-total word ratio.

In order to assess the significance of these features in prediction, we employed SHAP (SHAPley Additive Explanations) values to reverse-engineer the output of the predictive algorithm. This involved training the bot detection model on the aforementioned features from the three distinct classes of social bots. Subsequently, we evaluated the model's performance by testing it on the corresponding test set.

By utilizing SHAP values, we gain insights into the importance and contributions of each feature towards the model's predictions. This analysis enables us to understand the relative influence of different textual characteristics on the bot detection process. Through this approach, we can identify the key indicators and linguistic patterns that distinguish the various classes of social bots, further improving the effectiveness of our detection model.

4 Experimental results

4.1 Datasets

In this study, we use a dataset of Twitter platform that serves as a reference point for spambot detection (Cresci et al. 2017). This dataset includes two categories of accounts: genuine accounts and social bots. The genuine accounts managed by humans and social bots handeling by bots are divided into three subgroups: political bots (referred to as social spambots 1), financial bots (referred to as social spambots 2), and commercial bots (referred to as social spambots 3). To identify the political bots, we analyzed their activity of retweeting posts related to an Italian political candidate. The financial bots were identified as those responsible for spamming paid mobile apps. Finally, the commercial bots were identified as spammers promoting products on Amazon.com. We named these subgroups according to their main functions.

Here, we have extracted the conversations based on done replies on tweets by using the in_reply_to_status_id parameter available in the dataset.

The statistical information of the Cresci's dataset is summarized in Table 1. To prepare the dataset, we extracted Tweet-Retweet and Tweet-Reply relationships between users based on human-human and bot-bot interactions available in the dataset. Also, we used the midterm dataset that is filtered based on political tweets collected during the 2018 U.S. midterm elections (Yang et al. 2020).

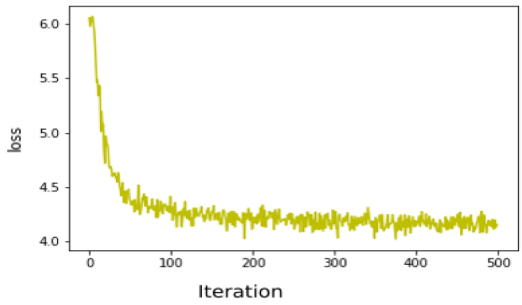
4.2 Scenario 1

To gain insight into the behavior of our models and identify areas where performance degradation occurs and finally propose solutions, we carried adversarial training.

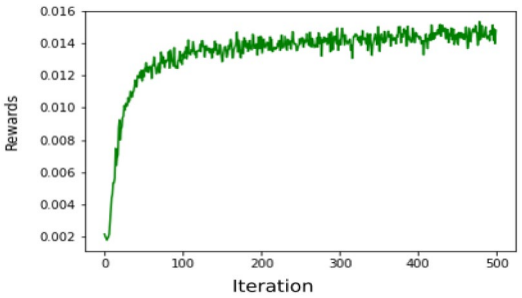
We used dialogue models to generate responses as a social bot, considering the context of ongoing conversations. On the other hand, we used a Contextual LSTM bot detection model to distinguish between machine-generated (social bots) and human-generated examples.

Table 1 Dataset - Cresci 2017 (Cresci et al. 2017) and Midtrerm 2018 (Yang et al. 2020)

Dataset	Label	Tweets	Conversations
Cresci2017	Human	39,264 (38,516 , 748)	16,967 (16593, 374)
	Bot1-Political	3810 (1,054, 2,756)	1778 (400, 1378)
	Bot2-Financial	932 (932, 0)	434 (434, 0)
	Bot3-Commercial	430 (176, 254)	200 (73, 127)
	Bot(Bot1+Bot2+Bot3)	5172 (2,162 , 3,010)	2,412 (907, 1,505)
midterm-2018	Human	8,092	
	Bot	42,446	



(a) Bot detection Loss



(b) Rewards received from the bot detector to the bot

Fig. 2 Training procedure of bot detection in the context of GAN

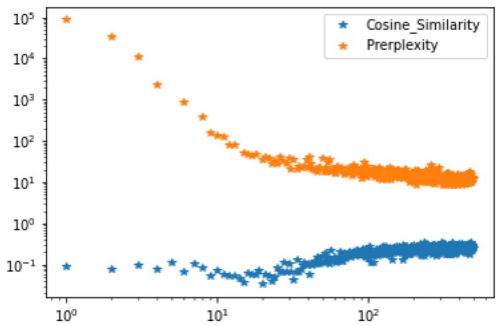


Fig. 3 Training procedure of bot in the context of GAN (Perplexity vs. Cosine similarity of generated samples to the real ones

In our adversarial scenario, our first goal was to evaluate the generative model's ability to produce samples that closely resembled real samples. We focused on evaluating how well the generated samples reproduce the features and patterns observed in the genuine data.

By comparing the generated samples to the real ones, we were able to determine the truth and realism of the generative model results based on GAN framework. This evaluation was important in determining the model's ability to reproduce the nuances and subtleties of the genuine data. It served as a fundamental benchmark for measuring the performance and accuracy of the generative model and provided valuable understanding into its capabilities. Figure 3 shows the result of this evaluation. The x-axis represents the number of training iterations for the generator model, while the y-axis represents the perplexity of the generated samples and the cosine similarity between the generated and actual samples. As can be seen in the figure, the cosine similarity between the generated samples and the real ones increases with the training of the bot model and the complexity of the bot decreases. This is to be expected as the generative model gets better at generating samples that can fool the discriminator model.

Our next goal is to quantify the fluctuations of the bot detector druring the adversarial training process. Figure 2 shows the loss and the rewards that the bot detector sends to the bot throughout the training process.

By analyzing the loss and reward curves, we represent the performance of bot detector model and its ability to discriminate human and bot-generated samples.

The observed decreasing loss of the bot detection model in the context of GAN demonstrates its improvement during training. As the model optimizes its parameters, it becomes more skilled at discriminating between samples generated by social bots and those generated by humans. This indicates a positive trend in the model's ability to accurately detect machine-generated samples.

According to the GAN concept, the increasing signal rewards received to the social bot suggests that the bot detection model is improving at identifying and classifying bot-generated samples. So, the improvement in the bot detection is a direct consequence of the training process in the GAN framework.

Here, we modeled a GAN-based game between bot and bot detection models. As shown, each of bot and bot detection models (traind on the bot dataset) are working right like the generative and discriminator models in the context of GAN.

Now, we evaluate the training speed of the bot and bot detection models based on their training process in the designed competition. Earlier in the evaluation of bot detector's performance 2 it can be observed that although the reward increases and the loss decreases, the plots in the later iterations exhibit smoother slopes. This trend suggests that the bot detection model encounters difficulty in extracting substantial information from the generated samples during these iterations. Moreover, the training loss of two models are compared in Fig. 5. Here, it is obvious that the training speed of the bot outperforms the performance of the bot detector model. In fact, the comparison of the loss values confirms the faster training speed of the generative model, indicating that it can outperform the bot detection model in terms of training efficiency.

Finding 1: In the GAN framework, a weak discriminator can be the reason of the other problems such as collapse mode, which can be solved by replacing autoencoders

(Pinaya et al. 2020), Energy-based GANs (Zhao et al. 2016) or a strong cost function (Che et al. 2020).

Since the behavior of the bot and the bot detector on the human-bot data is the same as the behavior of the generative and the discriminator models in the GAN context, we can use the solutions of the powerful discriminators proposed in the GAN framework for the bot detection method, too.

4.3 Scenario 2

In this part, we evaluate the bot detection model based on the attack examples poisoned to the dataset based on two different scenarios: attack examples are pinching from the human dataset or generating by a social bot.

4.3.1 Attack examples are pinching from the human's dataset

In social networks, bots can use deceptive tactics by spreading content written by humans and making other users believe that it comes from a real person. In this evaluation phase, we focus on evaluating the ability of our bot detection model to identify repeat bots that use this strategy on social media platforms.

Bots can use various strategies to mimic human behavior, as mentioned earlier. They can adapt their actions based on the performance of bot detection models and specifically target examples where they are less likely to be detected. In this experiment, we specifically select examples with a low probability of detection to evaluate the effectiveness of the model in detecting these deceptive behaviors.

The results presented in Fig. 4 show the probability distribution between three classes: human, bot, and attack examples. The attack examples are derived from the human class and placed in the bot class to simulate deceptive behavior. As expected, the detection probability differ among the three models trained on different types of bots during the training and testing process. Figure 4 shows that detecting attack examples within the bot class is more difficult for the model trained on commercial bots than for the models trained on political and financial bots. Furthermore, the model trained on financial bots shows greater difficulty in detecting attack examples than the model trained on political bots.

Finding 2: These results show that attack example detection varies in difficulty across different types of bots. Attack example detection confirm to be more difficult for the model trained on commercial bots, followed by the model trained on financial bots, while the model trained on political bots performs comparatively better.

4.3.2 Attack examples are generated human-like samples

To generate examples that mimic human behavior, we used a generative model trained in an adversarial game, treating it like a bot. These generated examples that resembled human-like interactions were then included in the bot class. A classifier was then used to evaluate the likelihood that these examples belonged to either the bot or human class.

The probability distribution of the bot, human, and attack examples is shown in Fig. 6. The results reveal interesting patterns. Models trained on bot examples from the political and financial domains have difficulty distinguishing attack examples from human examples, resulting in predictions that are close to the human class. However, when the generated examples are input to the model trained on commercial bot examples, it exhibits a higher probability of categorizing

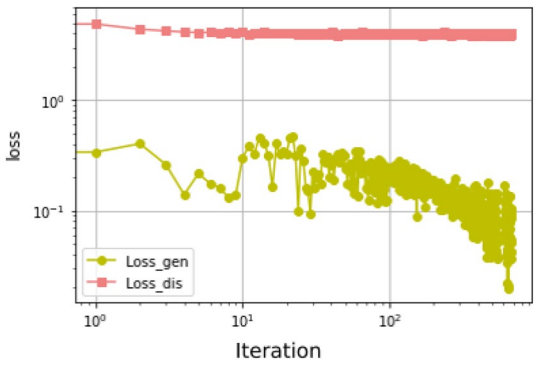
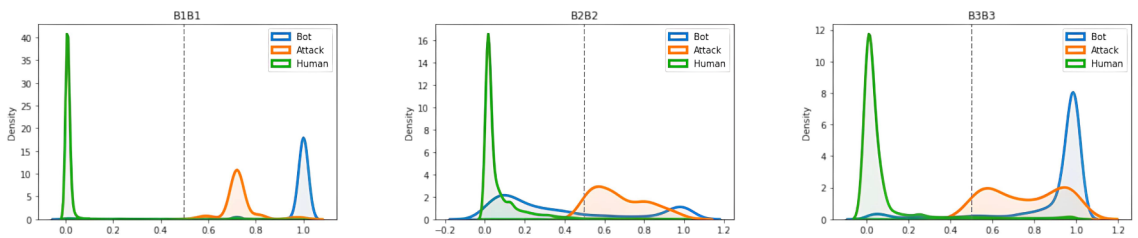
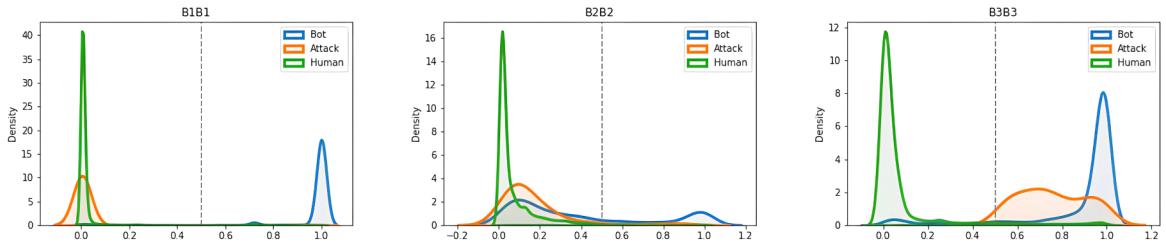


Fig. 5 Training speed of bot and bot detection in the context of GAN



(a) Bot1-Political (b) Bot2-Financial (c) Bot3-Commercial

Fig. 4 Probability distribution of three classes of social bots for attack examples selected from data



(a) Bot1-Political (b) Bot2-Financial (c) Bot3-Commercial

Fig. 6 Probability distribution of three classes of social bots for attack examples generated in adversarial games

them as belonging to the bot class. This indicates that the model trained on commercial social bots performs relatively worse than the models trained on political and financial bots.

Finding 3: These results suggest that the model trained on commercial bots has difficulty in accurately detecting and distinguishing attack samples from human samples, in contrast to the models trained on political and financial bots. The differences in performance suggest differences in the effectiveness of the models in detecting deceptive behaviors and highlight the need for further analysis.

4.4 Scenario 3

In this section, we will take a closer look at the textual characteristics of the three different classes of social bots and examine their differentiating features in more detail.

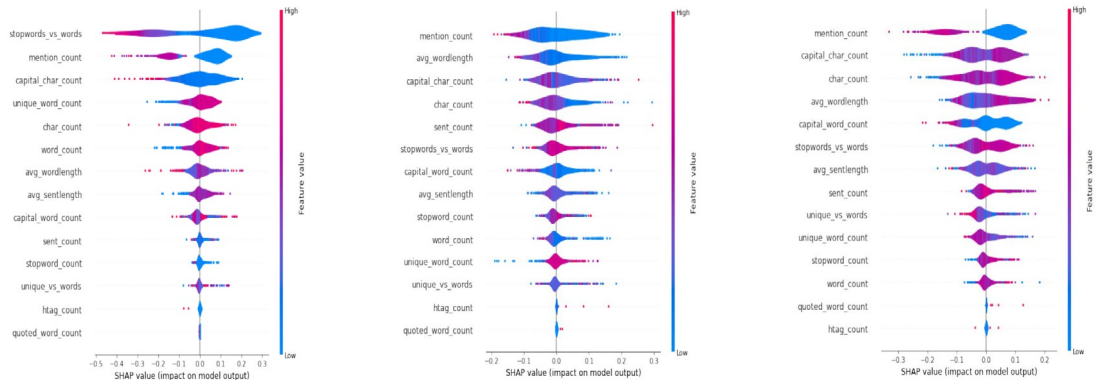
Figure 7 illustrates the relationship between the SHAP values and the prediction performance, with the horizontal axis representing the SHAP value and the color of the points indicating the relative values of the observations compared to those with higher or lower values.

The analysis shows that not all features have the same importance in prediction. For example, the number of mentions has a negative effect on prediction performance when its value is higher. Vice versa, lower mention count values have a positive effect on prediction performance for all three types of social bots.

In the case of political bots, the stopword_to_total word ratio feature displays a negative impact when its value is higher, while lower values have a positive impact on the prediction. On the other hand, this feature appears to be less significant for financial and commercial bots compared to other features. Consequently, it may be feasible to exclude this feature during the pre-processing step, given its relatively lower importance.

Furthermore, the results highlight the similarity between the features associated with financial and commercial bots, while political bots show distinct behavioral characteristics in these identified features.

Finding This analysis provides valuable insights into the importance and influence of different features on the bot detection model's predictions. By understanding the diverse impacts of these features, we can clarify our pre-processing steps and improve the overall accuracy and performance of the bot detection model, particularly when distinguishing between different types of social bots.



(a) Political bot (b) Financial Bot (c) Commercial Bot

Fig. 7 Explain bot detection model based on 14 NLP-based features

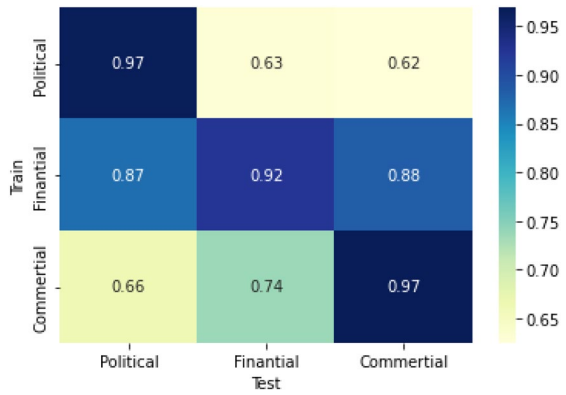


Fig. 8 Accuracy values for cross-data analysis

There have been some studies that aim to conduct a cross-domain analysis, in which the goal is to evaluate the generalization of a model by training it on a dataset from one domain and testing it on a dataset from a different domain(Yang et al. 2020).

To further assess the performance of a text-based classifier, we performed an evaluation where the classifier was trained on the human class and a specific social bot, and then tested on a human test set and another social bot. The outcomes of this evaluation are shown in Fig. 8.

The results demonstrate that the model exhibited poor performance when predicting the political social bot using a model trained on the commercial bot. This lack of accuracy can be attributed to the lowest correlation between the two domains. On the contrary, the model performed relatively similarly when predicting the commercial and financial bots using a model trained on the political bot, or when predicting the political bot using a model trained on the political bot itself.

This results indicate that the prediction of political bots using models trained on commercial and financial bots yields a higher accuracy. However, when the model trained on the financial bot is tested on the same dataset, its accuracy is lower compared to the other bot datasets. This discrepancy may be attributed to the fact that the model trained on the financial dataset exhibits better generalization capabilities, while the models trained on the commercial and political datasets are more prone to over-fitting on the training set, resulting in worse performance in different circumstances.

Also, these findings suggest that the classifier may be more effective at predicting political bots using models trained on commercial and financial bots. However, it may not generalize well to other scenarios or domains.

Finding 5: Here, our findings indicate that the Contextual-LSTM classifier may not be highly effective in accurately predicting social bots from different domains. The divergent characteristics and behaviors exhibited by social bots in different domains pose challenges for the classifier in effectively capturing and distinguishing their features.

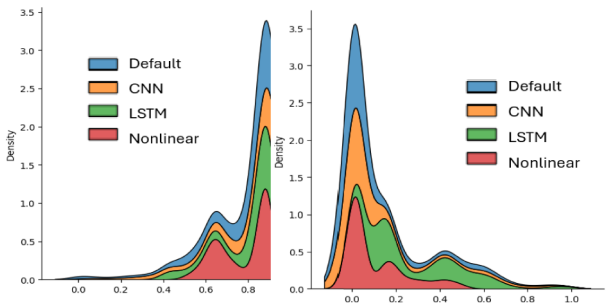
One challenge we face is finding well-labeled data for conversational bots that interact with humans. Such datasets are not widely available, which limits our ability to fully explore and validate our results. The scarcity of these datasets affects our capacity to train and test our models comprehensively on realistic human-bot interactions. As a result, our current findings may need further evaluation with more diverse and well-labeled data to provide a more complete picture of the model's performance.

4.5 Comparison

In this section, we have evaluated the effectiveness of four different approaches.

Since the generation of word embeddings from Bidirectional Encoder Representations from Transformers (BERT), a pre-trained language model, is an efficient technique, we used this method together with neural networks. Therefore, we used four models for comparison: CNN, LSTM, Nonlinear and Default. The default model relies exclusively on BERT for predictions.

Figure 9 shows the probability distribution of two classes, bot and human, for the Cresci dataset. It is noticeable that the probabilities for the bot class tend towards 1, while the probabilities for the human class tend towards 0. The default model has the highest distribution for both classes, followed by the CNN, LSTM and Nonlinear models in successive order. Table 2 shows the performance of these four models in two datasets, Cresci-2017 and Midterm-2018, confirming the previous results.



(a) Probability distribution for class bot (b) Probability distribution for class human

Fig. 9 Model Comparison - the probability distribution of two class bot and human in different models

Table 2 Dataset Comparison - Cresci 2017 (Cresci et al. 2017) Vs. Midtrerm 2018 (Yang et al. 2020) (Cresci/Midterm)

Model	Accuracy	AUCROC
Default	0.915/ 0.825	0.971/ 0.588
CNN	0.893/ 0.799	0.962/ 0.704
LSTM	0.819/ 0.825	0.902/ 0.522
NonLinear	0.875/ 0.825	0.951/ 0.523

5 Conclusions

This research employed a synthetic adversarial game to assess the effectiveness of text-based social bot detection methods in different scenarios. The focus was on evaluating the performance of generative models in generating attack examples that mimic human textual behavior. The findings revealed that these detection methods exhibit varying levels of strength across different types of social bots, with performance variations based on the specific domain and content generated by the bots.

In future works, the aim is to conduct more extensive bot detection models based on this works achievements. By gaining deeper insights into the behavioral patterns of social bots, the research aims to enhance the performance of detection models. This will involve refining the understanding of bot behavior and exploring novel approaches to improve detection accuracy and effectiveness.

By addressing the limitations identified in this study and further investigating the complexities of social bot detection, the research endeavors to advance the field and contribute to the development of more robust and reliable methods for detecting and combating social bots.

Acknowledgements Mostafa Salehi is supported by a grant from Institute for Research in Fundamental Science (IPM), Iran (No. CS1403-4-32)

References

Aghakhani H, Machiry A, Nilizadeh S, et al (2018) Detecting deceptive reviews using generative adversarial networks. In: 2018 IEEE Security and Privacy Workshops (SPW), IEEE, pp 89–95

Alarifi A, Alsaleh M, Al-Salman A (2016) Twitter turing test: identifying social machines. Inf Sci 372:332–346

Aljabri M, Zagrouba R, Shaahid A et al (2023) Machine learning-based social media bot detection: a comprehensive literature review. Soc Netw Anal Min 13(1):20

Arin E, Kutlu M (2023) Deep learning based social bot detection on twitter. IEEE Trans Inf Forensics Secur 18:1763–1772

Beskow DM, Carley KM (2019) Its all in a name: detecting and labeling bots by their name. Comput Math Organ Theor 25(1):24–35

Che T, Zhang R, Sohl-Dickstein J et al (2020) Your gan is secretly an energy-based model and you should use discriminator driven latent sampling. Advan Neural Inf Process Sys 33:12275–12287

Chen H, Liu X, Yin D et al (2017) A survey on dialogue systems: Recent advances and new frontiers. Acm Sigkdd Explor Newsl 19(2):25–35

Cresci S, Di Pietro R, Petrocchi M, et al (2017) The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In: Proceedings of the 26th International conference on world wide web companion, pp 963–972

Cresci S, Lillo F, Regoli D et al (2019) Cashtag piggybacking: uncovering spam and bot activity in stock microblogs on twitter. ACM Trans Web (TWEB) 13(2):1–27

De Nicola R, Petrocchi M, Pratelli M (2021) On the efficacy of old features for the detection of new bots. Inf Process Manag 58(6):102685

Dorri A, Abadi M, Dadfarnia M (2018) Socialbothunter: Botnet detection in twitter-like social networking services using semi-supervised collective classification. In: 2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), IEEE, pp 496–503

Ferrara E (2023) Social bot detection in the age of chatgpt: Challenges and opportunities. First Monday

Ferrara E, Varol O, Davis C et al (2016) The rise of social bots. Commun ACM 59(7):96–104

Gallotti R, Valle F, Castaldo N et al (2020) Assessing the risks of 'infodemics' in response to covid-19 epidemics. Nat Hum Behav 4(12):1285–1293

Goodfellow I, Pouget-Abadie J, Mirza M, et al (2014a) Generative adversarial nets. Advances in neural information processing systems 27

Goodfellow IJ, Shlens J, Szegedy C (2014b) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572

Hayawi K, Saha S, Masud MM et al (2023) Social media bot detection with deep learning methods: a systematic review. Neural Comput Appl 35(12):8903–8918

Heidari M, James Jr H, Uzuner O (2021) An empirical study of machine learning algorithms for social media bot detection. In: 2021 IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS), IEEE, pp 1–5

Jr SB, Campos GF, Tavares GM et al (2018) Detection of human, legitimate bot, and malicious bot in online social networks based on wavelets. ACM Transactions on Multimedia Computing, Communications, and Applications(TOMM) 14(1s):1–17

Kudugunta S, Ferrara E (2018) Deep neural networks for bot detection. Inf Sci 467:312–322

Li J, Galley M, Brockett C, et al (2015) A diversity-promoting objective function for neural conversation models. arXiv preprint arXiv:1510.03055

Najari S, Salehi M, Farahbakhsh R (2022) Ganbot: a gan-based framework for social bot detection. Soc Netw Anal Min 12(1):1–11

Nizzoli L, Tardelli S, Avvenuti M et al (2020) Charting the landscape of online cryptocurrency manipulation. IEEE Access 8:113230–113245

Orabi M, Mouheb D, Al Aghbari Z et al (2020) Detection of bots in social media: a systematic review. Inf Process Manag 57(4):102250

Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp 1532–1543

Pinaya WHL, Vieira S, Garcia-Dias R et al (2020) Autoencoders. In: Andrea M, Sandra V (eds) Machine learning. Elsevier, Amsterdam

Shehnepoor S, Togneri R, Liu W, Bennamoun M (2021) ScoreGAN: a fraud review detector based on regulated GAN with data augmentation. IEEE Trans Inf Forensics Secur 17:280–291

Varol O, Ferrara E, Davis C, et al (2017) Online human-bot interactions: Detection, estimation, and characterization. In: Proceedings of the international AAAI conference on web and social media, pp 280–289

Vaswani A, Shazeer N, Parmar N, et al (2017) Attention is all you need. Advances in neural information processing systems 30

Velayutham T, Tiwari PK (2017) Bot identification: Helping analysts for right data in twitter. In: 2017 3rd international conference on advances in computing, communication & automation (ICACCA) (fall), IEEE, pp 1–5

Wang AH (2010) Detecting spam bots in online social networking sites: a machine learning approach. In: Data and Applications Security and Privacy XXIV: 24th Annual IFIP WG 11.3 Working Conference, Rome, Italy, June 21-23, 2010. Proceedings 24, Springer, pp 335–342

Weiss K, Khoshgoftaar TM, Wang D (2016) A survey of transfer learning. J Big data 3(1):1–40

Wu Y, Fang Y, Shang S et al (2021) A novel framework for detecting social bots with deep neural networks and active learning. Knowl-Based Syst 211:106525

Xu Y, Zhao H, Zhang Z (2021) Topic-aware multi-turn dialogue modeling. In: Proceedings of the AAAI Conference on Artificial Intelligence, pp 14176–14184

Yang KC, Varol O, Hui PM, et al (2020) Scalable and generalizable social bot detection through data selection. In: Proceedings of the AAAI conference on artificial intelligence, pp 1096–1103

Yu L, Zhang W, Wang J, et al (2017) Seqgan: Sequence generative adversarial nets with policy gradient. In: Proceedings of the AAAI conference on artificial intelligence

Zago M, Nespoli P, Papamartzivanos D et al (2019) Screening out social bots interference: Are there any silver bullets? IEEE Commun Mag 57(8):98–104

Zhao J, Mathieu M, LeCun Y (2016) Energy-based generative adversarial network. arXiv preprint arXiv:1609.03126