Data & Knowledge Engineering 149 (2024) 102245

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

CALEB: A Conditional Adversarial Learning Framework to enhance bot detection

Ilias Dimitriadis *, George Dialektakis, Athena Vakali

Aristotle University of Thessaloniki, Informatics Department, University Campus, Thessaloniki, 54124, Greece

ARTICLE INFO

Keywords:
Online social networks
Social bot detection
Bot evolution
Adversarial machine learning
Conditional generative adversarial networks
Twitter

ABSTRACT

The high growth of Online Social Networks (OSNs) over the last few years has allowed automated accounts, known as social bots, to gain ground. As highlighted by other researchers, many of these bots have malicious purposes and tend to mimic human behavior, posing high-level security threats on OSN platforms. Moreover, recent studies have shown that social bots evolve over time by reforming and reinventing unforeseen and sophisticated characteristics, making them capable of evading the current machine learning state-of-the-art bot detection systems. This work is motivated by the critical need to establish adaptive bot detection methods in order to proactively capture unseen evolved bots towards healthier OSNs interactions. In contrast with most earlier supervised ML approaches which are limited by the inability to effectively detect new types of bots, this paper proposes CALEB, a robust end-to-end proactive framework based on the Conditional Generative Adversarial Network (CGAN) and its extension, Auxiliary Classifier GAN (AC-GAN), to simulate bot evolution by creating realistic synthetic instances of different bot types. These simulated evolved bots augment existing bot datasets and therefore enhance the detection of emerging generations of bots before they even appear. Furthermore, we show that our augmentation approach overpasses other earlier augmentation techniques which fail at simulating evolving bots. Extensive experimentation on well established public bot datasets, show that our approach offers a performance boost of up to 10% regarding the detection of new unseen bots. Finally, the use of the AC-GAN Discriminator as a bot detector, has outperformed former ML approaches, showcasing the efficiency of our end to end framework.

1. Introduction

Over the last decade, Online Social Networks (OSNs) have prevailed as the default communication, interaction, and information sharing platforms. Recent analysis reveals that there are 4.65 billion social media users around the world as of April 2022, equating to almost than 70 percent of the eligible global population [1]. A recent study [2] has shown that Twitter, one of the most 'active' OSNs with more than 436.4 million monthly active users, has a +90% growth of daily active users, confirming that OSNs have become an integral part of humans daily lives. However, the openness and easily accessible OSNs interfaces has triggered the rise of automated accounts, also known as social bots [3]. Such accounts are machine or human operated software, either benign or malicious, that tend to imitate human behavior by posting content and interacting with humans in order to achieve their goals [4]. Since they are automated, they operate much faster than human users, carrying out useful functions without the need for human supervision.

* Corresponding author.
E-mail addresses: idimitriad@csd.auth.gr (I. Dimitriadis), gdialekt@csd.auth.gr (G. Dialektakis), avakali@csd.auth.gr (A. Vakali).

https://doi.org/10.1016/j.datak.2023.102245
Received 8 December 2022; Received in revised form 30 October 2023; Accepted 11 November 2023
Available online 14 November 2023

Unfortunately, bots can also come in the form of malware, posing high-level security threats in OSN platforms [5]. Especially on Twitter, where humans are more open to expressing their opinions and views, social bots find a breeding ground to impact people's thoughts and mindsets. Spreading misinformation and fake news has become a major critical issue and it is widely recognized that public opinion is actively influenced in a meta-truth era at which online trust is at risk [6]. OSNs like Twitter, have already recognized the need to take measures against social bots. In accordance with a study from The Washington Post from May 2018, Twitter has identified almost 10 million bot accounts [7]. In addition, social bots are accountable for producing 35% of the content posted on Twitter, as discovered by another study [8]. Even more recent studies have shown that during the COVID pandemic, most of the Twitter content related to this topic was posted by bots [9,10]. Thus, the high degree of social bots influence on Twitter and their ability to tamper with our social ecosystems massively, is an actual threat to our societies and democracies which struggle with high polarization and hateful speech in online debates [11,12].

As social bots have increasingly shown malicious activity in OSNs, negatively affecting a large part of the global population, many earlier research studies have proposed techniques to detect bots [13]. These techniques mainly focused on supervised and unsupervised Machine Learning (ML) algorithms that aim at separating bot accounts from legitimate ones by several actions (eg. detecting fake content related to each account, examining the account profile itself, and investigating the network of the account) [14].

Even though many research studies have made a significant effort to detect social bots, most of them suffer from a significant weakness induced by the evolutionary nature of social bots, as demonstrated by a recent research survey [14]. Up to now, bot evolution and transformation remains a major obstacle in all bot detection methods. The evolution of bots also results in the appearance of new bot types, which adopt advanced features and cutting edge strategies that make their detection a struggling endeavor. Thus, as bots evolve and become smarter, they develop capabilities of hijacking and evading the state-of-the-art bot detection systems [15]. In addition, armies of malicious bots mimic human behavior and share more similar characteristics with legitimate human accounts [16], making them less distinguishable from actual humans. Finally, current state-of-the-art techniques follow a **reactive approach**, coming too late in advancing new bot detection solutions, only after evidence of new evolved bots [17]. In this context, research efforts always remain one step behind the malicious evolving bot development [18].

Recently, a new Adversarial Machine Learning (AML) approach for bot detection has gained ground, with techniques which aim to become **proactive** and capable of anticipating the evolution of bots and detecting their emerging generations [18]. Such methods involve a specific class of ML Neural Networks, namely Generative Adversarial Networks (GANs) [19]. GANs can be trained to produce artificial samples of evolved social bots representing and pre-cautiously capturing next generations of malicious bots.

This approach can offer insights about the vulnerabilities of existing bot detection systems, before even bot developers discover and effectively exploit their weaknesses [20]. Despite the emergence of Adversarial techniques in bot detection, this approach is still at an early experimental stage and has not been fully explored yet. For instance, the existing AML approaches focus only on discriminating bots from humans and produce synthetic samples of general purpose bots ignoring their different types that exist in OSN platforms. However, recent research has shown that several kinds of bots have emerged with different intentions and objectives, featuring distinctive characteristics [21]. Therefore, it is of vital importance to construct a detection system capable of classifying evolved bots of multiple types.

Attempting to address the above open problems referring to the multi-type detection of bots and their evolution, this paper proposes CALEB, a Conditional Adversarial Learning Framework to proactively enhance bot detection by creating synthetic bot instances of multiple types that simulate evolved bots. These simulated evolved bots are then used to augment existing bot datasets and develop a robust detection system towards new unseen multi-type bots. More specifically, the main contributions of this paper are as follows:

C1. **We propose a feature-based exploratory analysis of different bot types.**: This study takes a further step from co-authors previous work [21] and proceeds with an additional exploratory analysis of feature categories that validates and highlights the diversity between the different types of bots.

C2. **We introduce a novel proactive methodology for the detection of multi-type evolving bots**: Targeting to capture the evolutionary nature of social bots, we propose the use of Conditional Generative Adversarial Networks (CGAN) and Auxiliary Classifier Generative Adversarial Networks (AC-GAN), which are trained to generate synthetic bot samples that simulate evolved social bots. These artificial bot instances are used to enhance advanced ML classifiers in order to make them robust against future generations of bots. To the best of the authors' knowledge, this approach has not been presented in literature before.

C3. **We implement an end-to-end Generative Adversarial bot detection framework**: To reduce the training overhead, we propose the use of an end-to-end Generative Adversarial model, namely AC-GAN, capable of executing two major tasks: the generation of synthetic bot samples of different bot types and the detection of existing and future generations of social bots.

C4. **We validate the efficiency of our multi-type bot detection framework on various public bot datasets**. The results of the experimental evaluation reveal that the proposed methodology achieves comparable performance to other state-of-the-art techniques on class imbalance scenarios. Additionally, extensive experiments demonstrate that our approach effectively addresses the issue of bot evolution, achieving up to 10% performance boost in comparison to already established methods, that do not take into account the evolving nature of bots.

The remainder of this paper is organized as follows: Section 2 outlines earlier related work. Section 3 presents an exploratory analysis based on Twitter bot datasets. Section 4 justifies the proposed methodology for the detection of multi-type evolving bots and Section 5 illustrates our experimental evaluation along with the core results. Finally, Section 6 concludes this work and proposes some ideas for future research.

2. Related work

Identifying social bots presents a significant challenge due to their close resemblance to legitimate human accounts. Existing research primarily leans toward employing unsupervised and supervised methods, with the latter being more common, as demonstrated in previous studies [14]. Recently, a new machine learning approach, known as Adversarial Learning (AL) utilizing Generative Adversarial Networks (GANs), has gained prominence in the field of social bot detection, which we elaborate on in Section 2.3.

2.1. Unsupervised methods for bot detection

Unsupervised Learning is a type of machine learning algorithm which analyses unlabeled data and learns to extract latent patterns or groupings. In social bot detection, most unsupervised methods use clustering in order to discover clusters or, i.e., groups of social accounts based on similar characteristics. The goal is to find features that differentiate social accounts and form clusters that indicate whether an account is a human or a bot [22]. Compared to supervised methods, there is limited study on detecting social bots using Unsupervised Learning. For example, Ruan et al. (2015) [23] proposes the use of statistical inference and behavioral features to identify clusters of compromised accounts, marking them as social bots.

Another unsupervised approach, proposed by Cresci et al. (2020) [24], focuses on behavioral similarities to detect groups of users that present suspicious behavior. An online detection system, namely DeBot, proposed by Chavoshi et al. (2016) [25,26], groups accounts that present highly synchronous activity for a long duration and recognizes them as bots. DeBot does not require labeled data and uses a novel lag-sensitive hashing method to cluster user accounts into correlated sets. Similar to Debot, Mazza et al. (2019) [27] proposed an unsupervised framework, namely RTbust, which examines the retweeting activity of large groups of users and identifies temporal patterns. RTbust groups user accounts with similar retweeting behavior into the same cluster, labeling as bots those accounts that belong to clusters designated by malicious retweeting activity. In a recent study by Mannocci et al. [28], the use of multivariate time series data extends beyond simple binary classification of bots and humans. Instead, the study delves into a more detailed analysis, classifying bots into distinct bot types. It employs multidimensional temporal features extracted from user timelines to detect concentrated clusters of users exhibiting a high degree of similarity, a recognized indicator of automated activity.

Nevertheless, unsupervised methods encounter challenges in keeping pace with the dynamic characteristics of social bots, and they also grapple with scalability concerns due to potential computational resource intensiveness and limitations in efficiently managing extensive datasets or real-time bot detection scenarios.

2.2. Supervised methods for bot detection

As already mentioned, most existing approaches in the literature rely mainly on Machine Learning supervised methodologies [29–32]. Supervised methods mainly refer to classification, where a model is trained to distinguish between social bot accounts and legitimate human ones. All these methods explore the effectiveness of different feature sets combined with various classifiers, with the Random Forest classifier consistently proving dominant. Other efforts in bot detection move beyond simple classifiers to analyze the broader impact of social bots on Twitter. These efforts include examining political and marketing campaigns influenced by bots and addressing challenges related to evolving bot behavior, which often mimics human actions [33,34]. More advanced approaches address the issue on a multi-class level where algorithms are trained to discriminate the different types of social bots [12,21,35–38]. These methods use an ensemble approach of various classifiers, to finally determine whether an account is human or specific type of bot.

Nonetheless, the majority of these approaches have faced a lot of criticism towards their resilience, basically due to the fact that their results are totally dependent on the datasets that they have been trained on, [39]. Additionally, the limitations of Twitter bot detection datasets, including simplistic collection strategies and insufficient high-quality data, may lead to errors in classification, introducing biases in downstream analyses, underscoring the need for transparent data handling and labeling procedures to address these issues [40]. Finally, these methods are only effective on detecting the existing bots while failing to do so for their evolved versions. To address the challenge of bot evolution, where bots are adapted so that they can evade current detection systems, these early approaches develop improved systems only after evidence of new evolved bots have been observed, thus staying one step behind bot developers [17]. In an effort to address some of the issues above, a new family of ML algorithms has started to gain ground in the field of social bot detection, namely Adversarial Machine Learning (AML) [18,41].

2.3. Adversarial methods for bot detection

At their core, adversarial methods use Generative Adversarial Networks (GANs) to generate realistic synthetic bot representations that look like existing ones, augmenting the available datasets with advanced bot samples. For example, Bin Wu et al. (2019) [42], proposed a simple GAN to tackle the problem of imbalance between bot and legitimate samples presented in most public bot datasets. Specifically, a simple GAN was trained on a dataset [16,30] composed of 2433 samples (only 18% were bot accounts) to generate synthetic bot samples in order to enhance the original dataset. Then they trained a separate Neural Network on the augmented dataset to distinguish between bot and human accounts. The results showed that their GAN-based approach outperformed five state-of-the-art oversampling techniques. Another adversarial approach to solve the problem of class imbalance in bot data is presented

Table 1
Previous efforts on adversarial bot detection. CALEB matches all specs, while competitors miss one or more of the features.

Paper	Multiclass bot detection	Bot evolution	Discriminator as bot detector
C. Yin et al. (2018)	✗	✗	✓
Bin Wu et al. (2019)	✗	✗	✗
J Ma et al. (2019)	✗	✗	✓
Cresci et al. (2019)	✗	✓	–
STK Jan et al. (2020)	✗	✓	✓
Bin Wu et al. (2020)	✗	✗	✗
CALEB (our approach)	✓	✓	✓

in [7], where a Conditional Generative Adversarial Network (CGAN) is proposed that is able to control the specific class of samples being generated by feeding the GAN with auxiliary information about the class labels of the data [43]. This additional information comes from a Gaussian kernel density peak clustering algorithm which clusters the bot samples based on their features and assigns a different bot category per cluster. This process helps the CGAN create more realistic synthetic bot samples while eliminating the imbalances between and within bot class distributions. Experimental results revealed that the advanced CGAN outperformed state-of-the-art oversampling techniques such as Random Oversampling, SMOTE, and ADASYN, reaching an F1 score of 97.56%. Despite the success of the above works in dealing with class imbalance, they only focus on generating artificial bot samples that are quite similar to the pre-existing ones, leaving their detection models vulnerable to advanced and evolved bots. Moreover, they do not consider identifying and generating different types of bots; instead, they only approach bot detection as a binary classification problem.

The first approach towards dealing with bot evolution is performed by Cresci et al. (2019) [15] where they propose a proactive adversarial bot detection method using genetic algorithms. In detail, they develop a novel genetic algorithm, namely GenBot, which has the ability to create evolved Twitter spambots whose behavior looks like the behavior of legitimate accounts. GenBot is then combined with a digital DNA behavioral modeling technique [44] and produces synthetic evolved generations of bots by focusing on the sequences of actions of spambots and legitimate accounts on Twitter. The key point of this work is the ability to produce adversarial samples, i.e., synthetic bot accounts, that, through thorough experimentation, prove capable of evading state-of-the-art bot detection systems such as those in [44,45]. This work reveals that the reactive schema that most detection algorithms follow makes them extremely vulnerable to future generations of bots. Thus, research has to move in the direction of proactive strategies which promote detection models to a priori adapt, develop more sophisticated techniques and become more robust to the upcoming generations of evolved bots.

Another work that attempts to address the issue of bot evolution is presented in [46], where STK Jan et al. (2020) propose a GAN-based framework with two Generators for producing advanced artificial bot samples and show that this approach requires only 1% of labeled data to outperform existing methods. In this context, a distribution-aware data synthesis is proposed based on known legitimate accounts and limited bot ones. The main idea is to generate synthetic bot examples for the unoccupied regions in the feature space by differentiating "outlier regions" representing new bot variants and "clustered regions" representing legitimate users. For this reason, two Generators are used to create these two types of data in a different manner. The first Generator produces clustered samples by gradually decreasing the aggressiveness of the data synthesis as we get closer to the benign region. On the other hand, the other Generator is more aggressive to fill in the space. The Discriminator of the GAN is trained to distinguish real from artificial data and legitimate users from bots simultaneously. After training their GAN-based model on a real-world dataset containing network traffic data from Radware, they point out that the proposed model outperforms other state-of-the-art approaches such as an LSTM network and another GAN-based method, namely OCAN, needing only 1% labeled data. However, as they state in their paper, the proposed model cannot capture bots embedded in the benign region and legitimate users that behave quite differently from most other users, leaving their detection system vulnerable to those types of accounts.

There are also some other works that use AML, such as in [47] where a GAN-based framework is proposed to enhance the performance of previous botnet detection methods by generating synthetic botnet samples and the work in [48] where a text-based GAN is developed to detect rumors and fake news. However, both approaches belong to different bot domains and do not focus on Twitter bots, which is the main purpose of this paper.

Even though several attempts have been made using AML for Bot Detection, this field still remains considerably unexplored. For example, most Adversarial approaches [7,42,47,48] propose a GAN as an oversampling method to increase the number of bot samples without considering their evolutionary nature and adaptation that make them capable of evading the current state-of-the-art detection systems. Other Adversarial methodologies [15,46] that attempt to deal with the evolving behavior of bots, only focus on distinguishing bots from legitimate accounts without taking into consideration the different existing types of bots. Last but not least, there are only three research studies in the literature [46–48] that utilize the GAN's Discriminator as a bot detector, and they focus only on the binary classification problem. On the other hand, to detect bots, most GAN-based frameworks train additional ML classifiers to perform the final classification, adding additional training overhead in their pipeline.

In this work, we attempt to fill in the identified gaps, as can be seen in Table 1, by proposing CALEB, a Conditional Adversarial Learning Framework to enhance bot detection. More specifically, a Conditional Generative Adversarial Network and an Auxiliary Classifier GAN is proposed to generate simulated evolved bot examples and help towards establishing a robust multi-class bot

Table 2
Different bot types included in the utilized dataset along with the number of examples for each bot category.

Bot class	Description	Instances
Spam bot	Accounts that post spam content	17 071
Social bot	Bots that try to attract followers	11 653
Political bot	Bots that deal with politics	497
Cyborg	Human monitored bots	5891
Self-declared	Accounts that state they are bots	1198
Other bot	Other type of simple bots	2109
Human	Genuine human accounts	30 752
Total accounts	–	**69 171**

detection system capable of identifying the different existing types of social bots and their next generations. Moreover, we investigate whether synthetic data produced by GANs can boost the classification performance in multi-class imbalanced bot data. Finally, we evaluate the use of the Auxiliary Classifier GAN (AC-GAN), as an end-to-end bot detection system, that is able to both generate artificial samples of bots and detect their existing and evolved versions. To the best of our knowledge, this is the first approach using GANs towards multi-type bot detection.

3. Data exploratory analysis

It is well known that the performance of bot detection methods as well as generative models highly depends on the data used to train and evaluate the models. In this section, we discuss about the data we used in this work and we present an exploratory analysis regarding the features that describe the social accounts.

3.1. Dataset

Taking into consideration the universality and diversity, in this work we decided to use a dataset that was already created in [21]. This dataset consists of 24 different datasets, which most of them are accessible under the Data Repository section of Botometer [36–38], containing social accounts from Twitter while two of them are the result of a manual account search from Twitter. In the beginning, the original datasets were identified only by a binary label, human or bot. However, since the goal of this paper is to generate synthetic bot samples of multiple types, we followed the same bot type categorization as in [21], where bots are divided into six categories: spam bots, social bots, political bots, cyborgs, self-declared bots, and other bots. In addition, we also have the human accounts. The different types of bots along with the number of examples for each social account type are shown in Table 2.

Each social account in the dataset is represented by 310 features from five different categories. Specifically, there are 182 Content features, 58 Sentiment features, 29 Temporal features, 28 User features, and 13 Hashtag Correlation Features. Since the feature extraction process we followed has already been introduced in previous work [21], we are not going to thoroughly discuss how these features are selected and extracted. However, to give an intuition for the reader, we provide a short example of some feature categories below. For example, Content Features include text-relevant metrics to capture the source data semantics expressed in tweets, such as text size. User features reflect characteristics of the user's profile, such as number of followers, followers/friends ratio, etc.

3.2. Feature distribution exploratory analysis

As mentioned above, the dataset we used contains information about seven distinct account types. In this work, we decided to discard the other bots category, since it may contains bots from various bot types and therefore makes the analysis of the data more complicated. At this point, we wanted to examine how the features of each bot type differ compared to the others and how this distinction benefits the bot categorization we have considered. Therefore, we proceeded with an Exploratory Data Analysis which serves the purpose of evaluating the categorization of the different types of bots we have considered and is not related to the feature extraction process.

Since each feature category in our data consists of numerous features, it is infeasible to create plots to visualize their distribution. For that reason, we use Principal Component Analysis (PCA) [49], to reduce the dimensions of our dataset from 310 (which is the original number of features) to 5, one dimension for each feature category. In other words, each feature category is projected into one dimension. Therefore, each social account in our dataset now consists of five features. After applying dimensionality reduction, we construct Probability Distribution Function (PDF) plots for each feature category. Our goal is to examine and compare the distribution of each feature category between the different types of bots, as stated in (C1) paper contribution. To this end and due to the lack of space, we present an example of the distribution for the temporal, content and user features, as illustrated in Fig. 1.

Remark 1. The temporal and user features of political bots follow a distribution that **significantly deviates** from the distribution of the other bot types, as illustrated in the first and second sub-figure of Fig. 1, respectively.

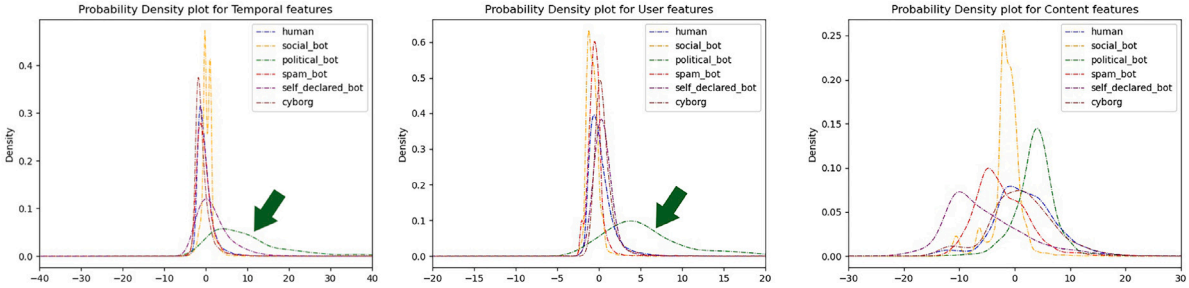

Fig. 1. Examples of probability distribution plots per feature category for different bot types.

In Fig. 1, we observe that the distribution of temporal and user features for political bots, respectively, differs from the other bots as it is centered around 5, while the curves for the other bot types are centered around zero, and has six times less height compared to the other curves. In terms of temporal features, this significant difference is somewhat expected as political bots diverge from the other classes since they are commonly more active during political events such as elections, etc, while the other bot types can be active on a daily basis.

Remark 2. The biggest divergence between the distribution curves is presented in the **content features**, as shown in the third part of Fig. 1.

Fig. 1 includes an example where the PDF distribution of the content features highly differs between the different bot types. It is obvious that none of the distributions is identical to another. On the other hand, for instance, in the user and temporal features, most bot types follow a similar distribution. This consideration leads us to our final key observation.

Remark 3. Content features is probably the **most important** feature category among the five we have considered for a classifier to discriminate between the different classes of bots.

As mentioned above, the PDF curves of the content features present the biggest difference between the different classes of social bots. This leads to the conclusion that this type of features contain more important information than the other categories, thus having more power to discriminate the different bot classes than features belonging to the other categories.

4. Methodology

In this section, we thoroughly describe our proposed methodology to address the unresolved issue of multi-type bot evolution.

4.1. Overall process

As outlined in Fig. 2, the proposed Bot Detection framework is composed of six distinct components:

1. **Data Collection & Storage**: The first step of this work requires collecting the necessary data. As already mentioned in Section 3.1, in this work we have used a dataset that was previously created in [21], which consists of publicly available data and manually collected data from Twitter representing social accounts.
2. **Preliminary Steps**: The following step in our pipeline refers to the preprocessing of the data, as well as the multi-type bot categorization. Since in this work we are interested to address the issue of bot evolution in multi-type bot data, we have followed the bot categorization that is presented in Table 2, as already discussed in Section 3.1.
3. **Feature Extraction & Engineering**: The final step before training our deep learning adversarial models is to extract the necessary features. As described in Section 3.1, each social account in our data is represented by 310 features that are divided in five categories: Temporal, Content, Sentiment, User, and Hashtag Correlation features.
4. **Synthetic Bot Data Generation**: Having collected and pre-processed the required bot data, we train two GAN models which we describe in Section 4.2, namely Conditional Generative Adversarial Network (CGAN) and Auxiliary Classifier-Generative Adversarial Network (AC-GAN), to generate realistic synthetic bot instances of multiple types.
5. **Data Augmentation**: The artificial bot data generated by GANs are then used to augment the original train and test sets, in a process we thoroughly describe in Section 5.4.
6. **Multi-type Bot Classification**: The last step in our pipeline requires training a ML classifier to perform the multi-type bot discrimination. For this purpose, we have decided to use a Random Forest classifier, as discussed in Section 5.1. In addition, instead of using an external ML classifier, we can use the Discriminator of AC-GAN, as we describe in Sections 4.2.2 and 5.5.

Next, we thoroughly describe CGAN and AC-GAN, which we use to generate synthetic instances of bots.

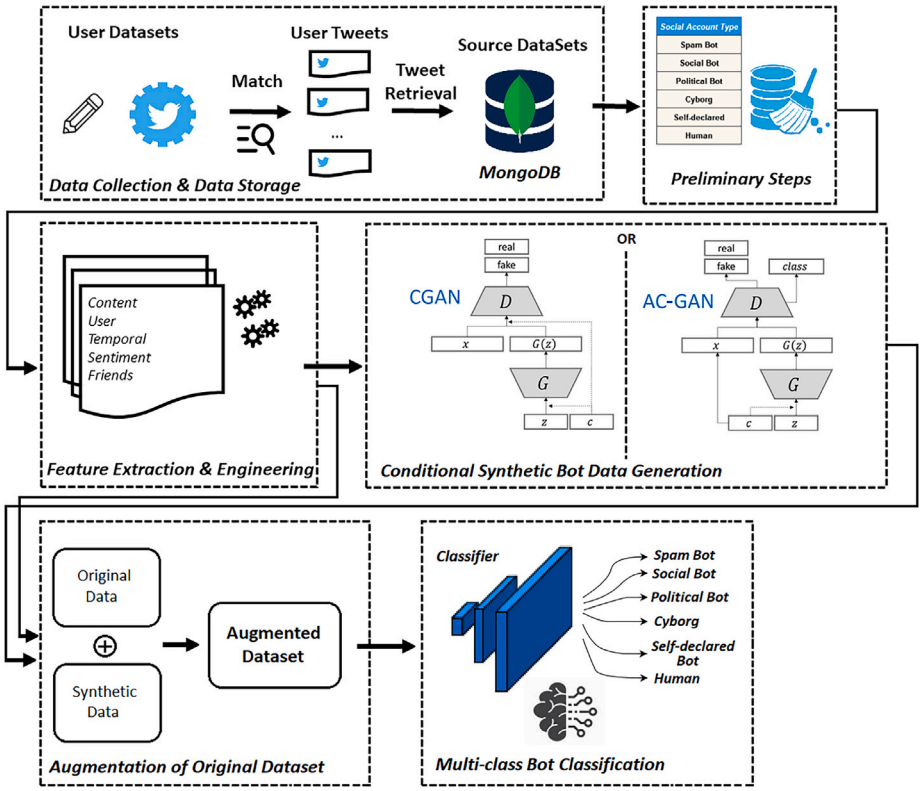

Fig. 2. Architecture pipeline of CALEB.

4.2. Synthetic bot data generation using GANs

Generative Adversarial Networks (GANs) have recently been introduced as a data augmentation technique, where a GAN is trained to generate realistic synthetic samples similar to original ones [50]. These artificially produced data can then be utilized to augment the original dataset and help ML algorithms achieve better performance. However, in the original GAN, there is no control on modes, i.e., classes, of the data being generated. Therefore, the idea behind using CGAN and AC-GAN to generate synthetic bot samples is that, in contrast with vanilla GAN, they offer the ability to create samples of specific classes. Besides the control over specific class generation, CGAN and AC-GAN were chosen instead of the Vanilla GAN since they have shown to provide a more stable training procedure [43,51].

4.2.1. Conditional generative adversarial networks

Conditional GAN, introduced in [43], is an extension of the vanilla GAN, as they both share almost the same model architecture as shown in Fig. 3. However, the main difference of the CGAN is that it incorporates additional information regarding the class labels in the input data. Specifically, both the Generator (G) and Discriminator (D) are conditioned on auxiliary information c by taking into account the class label for each sample in our training data. In a Conditional GAN (cGAN), an additional piece of information, often referred to as a condition or a label, plays a crucial role in guiding the generator (G) and discriminator (D) networks. Specifically, both G and D are conditioned on an integer value, which serves as a label indicating the bot type to which each data sample corresponds. This label acts as a vital condition for our bot generation process. The integer label, representing different bot types, is incorporated into our cGAN framework to ensure that the generated bot instances exhibit the characteristics and behaviors associated with the specified category. The information for the class labels is fed to both G and D as an additional Embedding input layer. In G, the Embedding layer with the class labels is concatenated with the prior input noise $p_z(z)$. In D, the Embedding layer with the class labels is concatenated with the data samples, either real or fake, and is given to the input layer. Similar to the vanilla GAN, G and D are both trained simultaneously. To elaborate on the role of the discriminator network D, it plays a crucial part in guiding the training of the generator network G. D is responsible for distinguishing between real bot samples and fake (generated) bot samples. This adversarial process, where G and D are in a constant competition, forces G to improve its capability to create more realistic and accurate bot instances. In essence, the discriminator network acts as a critical feedback mechanism for the generator. As D becomes more proficient at distinguishing between real and fake data, G is pushed to enhance its generation process. This iterative competition between G and D in a game-theoretic fashion results in the generation of bot instances that progressively approach the realism of real data.

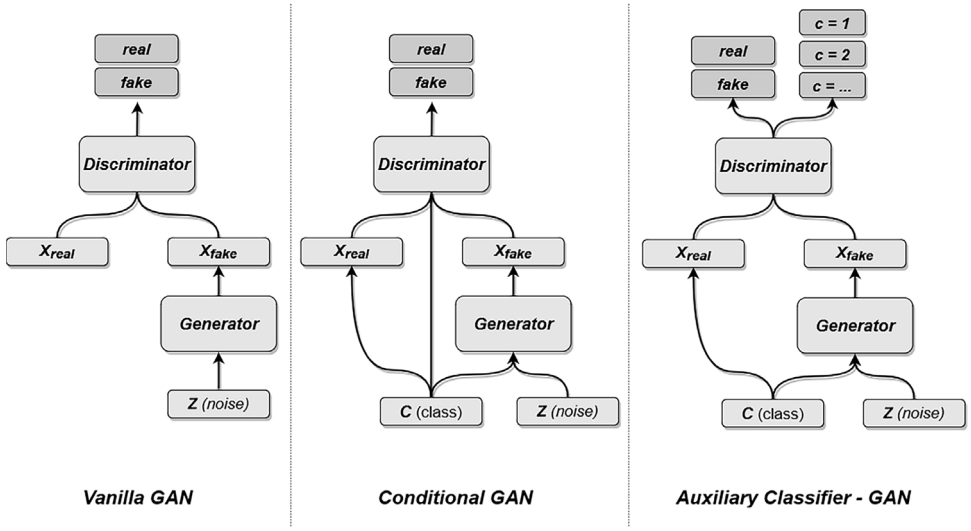

Fig. 3. GAN architectures comparison.

The parameters of G are adjusted to minimize $\log(1 - D(G(z \mid c)))$, and the parameters of D are adjusted to minimize $\log(D(X \mid c))$ as if they are following the two-player min–max game with value function $V(D, G)$:

$$\begin{aligned} \min_G \max_D V(D,G) &= \mathbb{E}_{x \sim p_{\text{data}}\ (x)}[\log D(x \mid c)] \\ &+ \mathbb{E}_{z \sim p_x(z)}[\log(1 - D(G(z \mid c)))] \end{aligned} \tag{1}$$

4.2.2. Auxiliary classifier-Generative adversarial networks

The Auxiliary Classifier-Generative Adversarial Network (AC-GAN) is an extension of the CGAN, introduced by A. Odena et al. [51]. As presented in Fig. 3, the main difference between CGAN and AC-GAN is in the discriminator model, which in the case of the AC-GAN, it is not fed with the class label of our data. **Instead of receiving the class label as input, the discriminator is trained to predict it**. Therefore, D has two outputs, one for the probability distribution over sources (training dataset or generator) and one for the probability distribution over the class labels. In other words, D simultaneously performs two tasks: determining the authenticity of a bot sample and assigning it to the appropriate class. This dual role of D has a profound impact on the training process. It not only helps identify fake samples but also guides the Generator (G) to produce more realistic and accurate bot examples. It is worth noting that the Generator of AC-GAN, much like the CGAN, is also conditioned on the class label of each data sample, corresponding to a specific bot type. This conditioning ensures that the generated bots align closely with the desired attributes and behaviors of the respective bot categories. By considering these conditions, we aim to create a diverse and representative set of bot instances, each tailored to mimic the characteristics and behaviors of different bot types. The selection of these conditions is driven by the need to produce bot instances that accurately reflect the multifaceted nature of bots on Twitter and to fulfill specific criteria in our study.

The objective function of the AC-GAN is composed of two parts: the log-likelihood of the correct source, L_S, and the log-likelihood of the correct class, L_C.

$$\begin{aligned} L_S &= E\left[\log P\left(S = \text{ real } \mid X_{\text{real}}\right)\right] \\ &+ E\left[\log P\left(S = \text{ fake } \mid X_{\text{fake}}\right)\right] \end{aligned} \tag{2}$$

$$L_C = E\left[\log P\left(C = c \mid X_{\text{real}}\right)\right] + E\left[\log P\left(C = c \mid X_{\text{fake}}\right)\right] \tag{3}$$

D is trained to maximize $L_S + L_C$ since its goal is to predict both the source of data and the class they belong to with the highest credibility possible. On the other hand, G is trained to maximize $L_C - L_S$, since its objective is to fool the discriminator into believing that generated data come from the training set while helping D map data to labels.

The most significant benefit of AC-GAN is that the discriminator can now be used as a classifier to detect the different types of social bots. Therefore, AC-GAN serves two purposes. First, we can use the generator of the AC-GAN to create realistic synthetic bot instances. Secondly, we can utilize the discriminator of the model for multi-class bot classification, reducing the training overhead that applies when training additional ML classifiers on the synthetic data for bot detection, in a process we illustrate in Section 5.5.

5. Experimental evaluation

In this section, we describe the experimental procedure we followed, and we highlight the key observations and the main findings.

Table 3
Class imbalance results using Random Forest with different augmentation techniques.

Augmentation technique	Accuracy	Precision	F1 score	Recall	G-Mean
Original	0.8903	0.9057	0.8762	0.8520	0.9095
ADASYN	0.8866	0.8570	0.8736	0.8922	0.9315
SMOTE-ENN	0.8537	0.7940	0.8370	0.8996	0.9327
CGAN 2 : 1	0.8888	0.9069	0.8788	0.8518	0.9094

5.1. Initial setup

Following the state-of-the-art hyper-parameter tuning of CGAN and AC-GAN, our GAN models are composed of feed-forward neural networks for the Generator and Discriminator. We used a noise vector of size 128, and the size of the Embedding layer was set to 6 to match the number of classes in our data. The loss function for both networks was constructed using Binary Cross-Entropy loss. The two models were trained using stochastic gradient descent with the Adam optimizer and mini-batches of size 512 and a learning rate of 0.0002. We chose to train our models for 300 epochs, since the loss did not show to decrease any further after that point. The implementation was made using the PyTorch open-source machine learning library in Python. Finally, we decided to use a Random Forest (RF) classifier with default parameters to perform the classification, since it has been proven by numerous works to be one of the most dominant classifiers for bot detection [52–55]. For all experiments we considered a 75%–25% train–test split.

5.2. Handling class imbalance

Before proceeding with the bot evolution results, we wanted to address the problem of class imbalance in our data, as shown in Table 2. To this end, we used CGAN and AC-GAN to generate synthetic samples using the expansion multiple to augment the original training dataset. The expansion multiple is defined as:

$$\varphi = a : b \tag{4}$$

where b is the number of social accounts per class in the training data, and a is the number of samples generated by the CGAN or AC-GAN. Drawing intuition from [7], we have decided to apply $\varphi = 2 : 1$ in all classes, meaning that for each sample in the training set, we generate two synthetic examples. We trained RF on the CGAN augmented data and evaluated its performance on o hold-out test set. We compared CGAN augmented data to two other state-of-the-art oversampling techniques, namely ADASYN [56] and SMOTE-ENN [57]. In addition, we report the results when no imbalance handling technique is used, as presented in Table 3.

Remark 4. Overall, we observe that ADASYN offers the best performance while our proposed CGAN method comes second best. In addition, we notice that Random Forest maintains a remarkably high performance even when no imbalance technique is used.

The above remark highlights that even though class imbalance is present in our data, no augmentation technique is necessary in order to maintain a high classification performance. Therefore, we proceed with the following experiments without addressing the class-imbalance in our data.

5.3. Synthetic data evaluation

Since the primary goal of this work is to tackle the challenge of bot evolution, we need to examine the quality of the synthetic data we are going to use for training and testing RF, and inspect the similarity between the original data and the artificial ones produced either by our Adversarial models (CGAN and AC-GAN) or ADASYN. For this reason, in Fig. 4 we present the similarity between the original data and different sets of synthetic data (CGAN, AC-GAN, and ADASYN), using two synthetic data evaluation metrics from the Synthetic Data Vault [58], the Kolmogorov–Smirnov (KS) test and the continuous Kullback–Leibler (KL) Divergence metric. The choice for the KS-test metric was made since it uses the two-sample Kolmogorov–Smirnov test to compare the distributions of continuous columns using the empirical CDF. The output for each column is 1 minus the KS Test D statistic, which indicates the maximum distance between the expected CDF and the observed CDF values [59]. As far as it concerns the KL divergence, the choice was made since it is the most familiar type of metric that is used in statistics to measure the similarity between two distributions [60].

Remark 5. We observe that ADASYN creates copies of the original data since its synthetic data have almost identical distribution to the original. On the other hand, CGAN and AC-GAN generate artificial data that feature some variety. In this way, the novel **synthetic GAN data cover a broader range of bots that can simulate different types of evolving bots**.

This remark points out the fact that not every augmentation technique can be used to simulate evolving bots, since it must create synthetic bot instances that introduce some variation compared to the original data. Our proposed approach achieves this variation between original and artificially made data emphasizing this work's (C2) contribution.

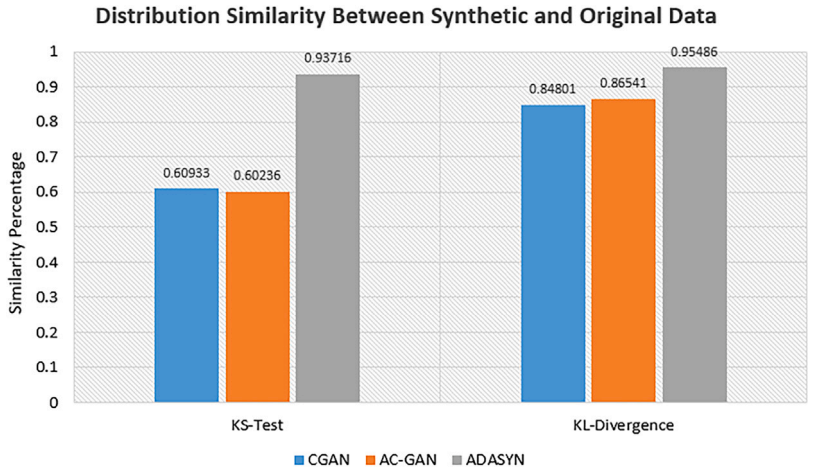

Fig. 4. Distribution similarity between original and synthetic data of different techniques.

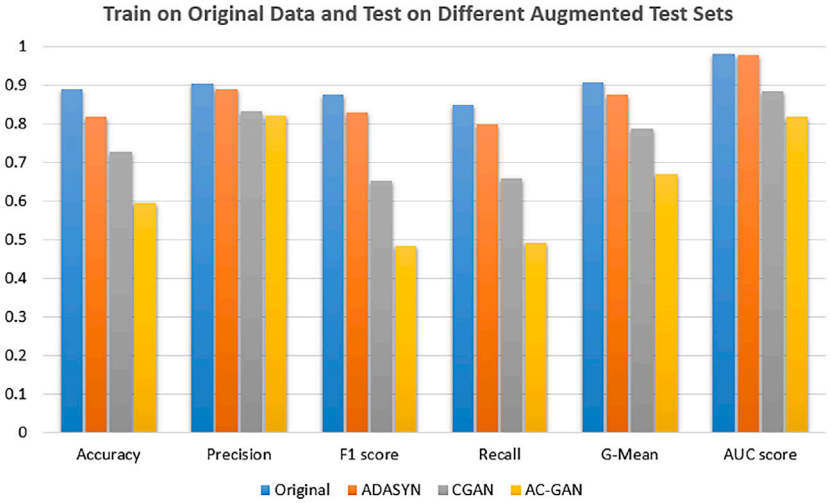

Fig. 5. Evaluation on different augmented test sets by training Random Forest only original data.

5.4. Bot evolution results

The first investigation we are interested in is the ability of a classifier trained only on existing bot data to detect future generations of bots. To this end, we simulate the evolution of bots by creating realistic synthetic bot instances using three different technique, two GAN models as described in Section 4, and ADASYN. At this point, we should mention that SMOTE-ENN is omitted for the rest of the experiments, since it showed inferior performance compared to ADASYN, as discussed in Section 5.2. We then induce the synthetic data into the original test set, constructing different augmented test sets based on the technique that was used to generate the artificial data. Finally, we train a Random Forest classifier using only the original training data and we evaluate the model on the three augmented test sets, as described above. In addition, we report the results when we evaluate the model on the pure original test set. The results of this process are presented in Fig. 5.

As can be observed, the classification performance of RF is greatly decreased when the original test set is augmented with synthetic data, i.e., when simulated bots make their appearance in the test set. ADASYN synthetic data do not diminish RF's performance by a big margin, which validates our observations as stated in Section 5.3. Therefore, an ML classifier trained only on the original data can still classify ADASYN's data with acceptable performance, which is above 80% in terms of all the evaluation metrics. On the other hand, when we induce CGAN or AC-GAN synthetic data into the original test data, we observe a **significant performance decrease** across all evaluation metrics, with the biggest impact presented in F1-score and Recall which fall under 50%. This poor performance suggests that RF is not able to classify the simulated evolved bots and requires additional training information to be able to succeed.

Towards this end, we consider three different augmentation techniques for the training data, along with the pure original training set, as follows:

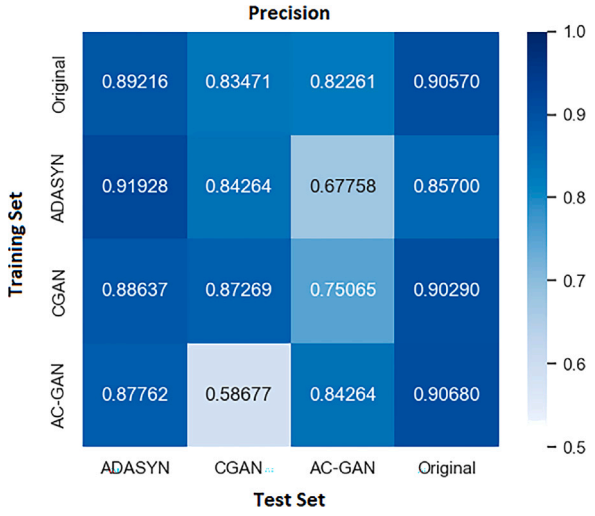

Fig. 6. Precision.

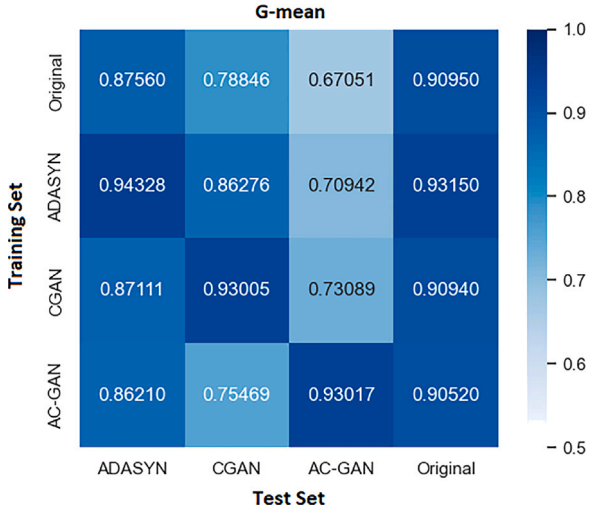

Fig. 7. G-mean.

- ad_1: original data augmented with **AC-GAN** synthetic data
- ad_2: original data augmented with **CGAN** synthetic data
- ad_3: original data augmented with **ADASYN** synthetic data
- ad_4: pure original data

The above training sets are illustrated on the y axis in Figs. 6 and 7. It is worth mentioning that the 2:1 expansion multiple was used to generate synthetic data from CGAN and AC-GAN, since it proved to be the best technique during the class imbalance experimentation. In Fig. 6, we present the Precision and in Fig. 7 the G-mean that RF obtains under different combinations of training and test data. It is obvious that RF achieves the best performance when trained and tested on the same type of augmented data. However, RF underperforms especially when tested on CGAN and AC-GAN augmented data while it has been previously trained only on ad_3 or ad_4. On the contrary, when we train RF on ad_1 or ad_2 and test on ADASYN augmented data, RF achieves relatively high performance, as shown in the Figures above.

So far, the experimental evaluation on different combinations of augmented train and test sets has led to the following observations:

Remark 6. ADASYN's synthetic data are not useful for simulating evolving social bots, since they include copies of the original data and can be easily classified by RF trained only on the original data.

Table 4
Random Forest performance trained on an older bot dataset ([61] 2011) and its augmented variations and evaluated on newer datasets ([62] 2017, [30] 2017, [63] 2018).

Test data	Train data	Accuracy	Precision	F1 score	Recall	G-Mean
Gilani (2017)	Caverlee (2011)	0.59298	0.58490	0.53621	0.55783	0.48260
	Augmented with ADASYN	0.58802	0.58356	0.51006	0.54654	0.43546
	Augmented with CGAN	0.58937	0.58164	0.52402	0.55153	0.55153
	Augmented with AC-GAN	**0.61279**	**0.61620**	**0.55588**	**0.57693**	**0.50251**
Varol (2017)	Caverlee (2011)	0.79186	0.76903	0.76105	0.75521	0.74686
	Augmented with ADASYN	0.78465	0.76638	0.74430	0.73317	0.71608
	Augmented with CGAN	0.79650	0.77445	0.76647	0.76059	0.76059
	Augmented with AC-GAN	**0.80732**	**0.78533**	**0.78169**	**0.77856**	**0.76927**
Cresci_Stock (2018)	Caverlee (2011)	0.57139	0.58350	0.56933	0.58011	0.56885
	Augmented with ADASYN	0.59539	0.60676	0.59402	0.60339	0.59429
	Augmented with CGAN	0.59818	0.60367	0.59807	0.60301	0.60301
	Augmented with AC-GAN	**0.62840**	**0.63545**	**0.62816**	**0.63405**	**0.61708**

Remark 7. CGAN and AC-GAN synthetic data greatly decrease RF's performance when it is trained on other training sets, with AC-GAN synthetic data being the most difficult to classify correctly.

Up to now, to validate the efficiency of our proposed approach we have used simulated evolved bots. However, this is not the ideal way to evaluate our method, since we are not fully convinced whether our GAN-based methodology accurately simulates bot evolution. To this end, we have decided to proceed with further experimentation based on a real world scenario, as described below.

Real Evolved Bot Data Experimentation:

To further verify that our proposed adversarial methodology can effectively simulate bot evolution and be used proactively to detect future generations of bots, we considered a real-world example using binary bot data. Initially, we trained a Random Forest classifier on an old public dataset (Caverlee 2011 [61]). We then evaluated this model on three more recent datasets (Gilani 2017 [62], Varol 2017 [30], and Cresci_Stock 2018 [63]), that include evolved bot instances. We proceeded by augmenting the original data with CGAN and AC-GAN using the 2:1 expansion multiple as in the above experiments. The idea behind this experiment is that the more recent datasets contain newer evolved bots that do not exist in the older dataset.

As can be seen in Table 4, augmenting the original training data with AC-GAN always boosts the performance of Random Forest, regardless of which one of the three newer datasets is our test set. On the other hand, augmenting with CGAN does not always increase the performance of the classifier and is always inferior to AC-GAN. At this point, we should mention that the absolute scores are not so crucial since the test datasets contain different types of bots, and we only focus on binary classification. On the contrary, we are interested in the performance boost that our proposed methodology offers. For instance, we obtain a **performance boost of almost 10%** when we evaluate RF on the Cresci_Stock [63] dataset which is the most recent among the ones we have considered for this experiment. This reveals that our adversarial methodology can effectively generate realistic synthetic bot samples that simulate evolving bots and help towards the detection of their future generations before they even emerge, once again highlighting this paper's (C2 and C4) contributions. In addition, in Table 4, we include the results when we augment the original training data with ADASYN. At this point, we should mention that even though five evaluation metrics are being presented, we emphasize on the Precision score since this metric penalizes False Positives. This is of vital importance in our task because in real-world applications it is very important to correctly classify human accounts as legitimate ones, and thus we are interested in reducing the number of False Positives, i.e. the number of human accounts incorrectly classified as bots.

The results show that not only ADASYN does not offer any improvement in the performance but in many cases it degrades it, confirming that ADASYN is not a suitable method for simulating evolving bots and showcasing that not all techniques that generate synthetic data can simulate bot evolution.

5.5. Evaluating AC-GAN as a bot detector

So far, throughout our experimental evaluation, we have considered only Random Forest as our bot detector to perform the classification of the different types of bots. However, this approach requires to first construct the augmented training datasets and then train additional ML classifiers, such as RF, to perform the classification, adding extra overhead in our pipeline. An alternative approach is to directly use the already trained Discriminator network of the AC-GAN for multi-class classification.

In Fig. 8, we present a performance comparison between RF and AC-GAN's Discriminator on mixed augmented data, which consist of original data, CGAN and AC-GAN synthetic data. The idea behind using a mixed augmented dataset is that we want to provide a fair evaluation of the classification models, since evaluating the discriminator of AC-GAN only on AC-GAN augmented data would obviously provide better results than RF. In this experiment, RF is trained once with CGAN augmented data, denoted by RF_CGAN, and once with AC-GAN augmented data, denoted by RF_AC-GAN. On the other hand, AC-GAN only uses the original training data to train both the Generator and the Discriminator. As can be observed, AC-GAN's Discriminator outperforms both RF with CGAN and AC-GAN data.

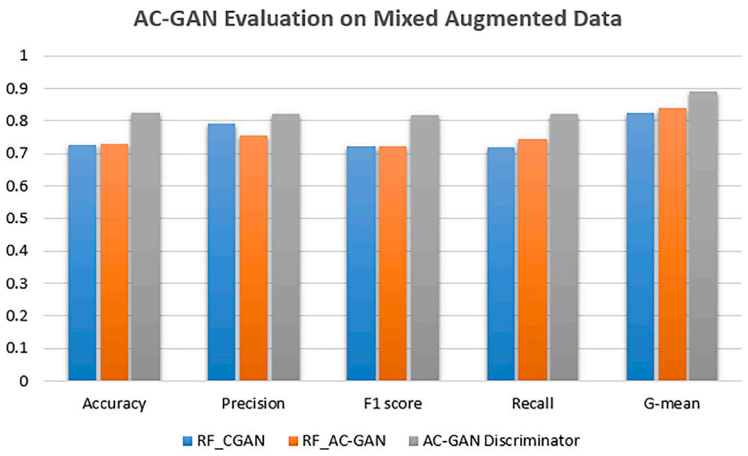

Fig. 8. Performance comparison between Random Forest (RF) and discriminator of AC-GAN on mixed augmented data.

Remark 8. The already trained Discriminator of AC-GAN can be used for the classification of multi-type evolved bots, omitting the need for additional ML classifiers. Therefore, AC-GAN composes an end-to-end bot detection framework which is **robust against evolving bots**, serving two purposes, generating synthetic bot data and successfully classifying existing and evolving social bots.

The above results and observation justify the most important contribution of this paper (C3), since to the best of our knowledge, this is the first time a GAN is used for classification in the bot detection domain.

6. Conclusion and future work

In this paper, we propose CALEB, a Conditional Adversarial Learning Framework to proactively detect multi-type evolving bots in Online Social Networks. In this context, we employed two GAN models, namely CGAN and AC-GAN, which were trained to create realistic synthetic bot instances of multiple types. The artificial GAN data represented evolved versions of bots and were used to augment the existing bot datasets in order to construct a robust ML classifier against future generations of bots.

Results showed that CALEB can effectively simulate evolving bots and help ML models detect future generations of bots with better performance compared to previous works. Moreover, our experimental analysis showed that other augmentation techniques that are widely used in class imbalance problems, such as ADASYN, are not suitable for simulating evolving bots, since they create synthetic data that are almost identical to the original one. Finally, we evaluated the Discriminator of AC-GAN as a bot detector, which showed to outperform Random Forest, revealing that there is no need to train additional classifiers to perform the multi-type detection of evolving bots.

Future work may focus on creating synthetic bot samples that present specific modified features using Controllable GANs [64], by leveraging the latent space disentanglement properties of the GAN. In this way, we may be able to simulate evolved bots in a more accurate way. In addition, an interesting idea for the future is to construct a robust set of language-agnostic features to overcome the limitation that is presented in most existing public bot datasets by non-English Twitter content.

CRediT authorship contribution statement

Ilias Dimitriadis: Conceptualization, Methodology, Data curation, Investigation, Writing – original draft, Writing – review & editing. **George Dialektakis:** Writing – original draft, Investigation, Validation, Data curation, Software, Visualization. **Athena Vakali:** Writing – review & editing, Supervision, Funding acquisition, Project administration, Resources.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Ilias Dimitriadis reports financial support was provided by Operational Program Competitiveness, Entrepreneurship and Innovation under the call RESEARCH-CREATE-INNOVATE co-financed by Greece and European Union (project T2EDK-03898).

Acknowledgment

All authors approved the version of the manuscript to be published.

Funding

This research is co-financed by Greece and European Union through the Operational Program Competitiveness, Entrepreneurship and Innovation under the call RESEARCH-CREATE-INNOVATE (project T2EDK-03898).

References

[1] Global social media Stats, 2021, retrieved from: https://datareportal.com/social-media-users.
[2] Essential Twitter stats for 2021, 2021, retrieved from: https://datareportal.com/essential-twitter-stats.
[3] E. Ferrara, O. Varol, C. Davis, F. Menczer, A. Flammini, The rise of social bots, Commun. ACM 59 (7) (2016) 96–104.
[4] S. Stieglitz, F. Brachten, B. Ross, A.-K. Jung, Do social bots dream of electric sheep? A categorisation of social media bot accounts, 2017, arXiv preprint arXiv:1710.04044.
[5] V. Luckerson, Can Twitter solve its big, bad bot problem?, 2018, Available at: https://www.theringer.com/tech/2018/3/8/17093982/twitter-bot-problem.
[6] C.A. Cassa, R. Chunara, K. Mandl, J.S. Brownstein, Twitter as a sentinel in emergency situations: lessons from the Boston marathon explosions, PLoS Curr. 5 (2013).
[7] B. Wu, L. Liu, Y. Yang, K. Zheng, X. Wang, Using improved conditional generative adversarial networks to detect social bots on Twitter, IEEE Access 8 (2020) 36664–36680.
[8] N. Abokhodair, D. Yoo, D.W. McDonald, Dissecting a social botnet: Growth, content and influence in Twitter, in: Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing, 2015, pp. 839–851.
[9] V.A. Young, Nearly half of the Twitter accounts discussing 'Reopening America' May be bots, 2020, URL https://www.cmu.edu/news/stories/archives/2020/may/twitter-bot-campaign.html.
[10] E. Ferrara, # Covid-19 on twitter: Bots, conspiracies, and social media activism, 2020, arXiv preprint arXiv:2004.09531.
[11] D.A. Broniatowski, A.M. Jamison, S. Qi, L. AlKulaib, T. Chen, A. Benton, S.C. Quinn, M. Dredze, Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate, Am. J. Public Health 108 (10) (2018) 1378–1384.
[12] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, A. Vakali, Mean birds: Detecting aggression and bullying on twitter, in: Proceedings of the 2017 ACM on Web Science Conference, 2017, pp. 13–22.
[13] V.S. Subrahmanian, A. Azaria, S. Durst, V. Kagan, A. Galstyan, K. Lerman, L. Zhu, E. Ferrara, A. Flammini, F. Menczer, The DARPA Twitter bot challenge, Computer 49 (6) (2016) 38–46.
[14] S. Cresci, A decade of social bot detection, Commun. ACM 63 (10) (2020) 72–83.
[15] S. Cresci, M. Petrocchi, A. Spognardi, S. Tognazzi, Better safe than sorry: An adversarial approach to improve social bot detection, in: Proceedings of the 10th ACM Conference on Web Science, 2019, pp. 47–56.
[16] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, 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, 2017, pp. 963–972.
[17] S. Cresci, M. Petrocchi, A. Spognardi, S. Tognazzi, From reaction to proaction: Unexplored ways to the detection of evolving spambots, in: Companion Proceedings of the the Web Conference 2018, 2018, pp. 1469–1470.
[18] S. Cresci, M. Petrocchi, A. Spognardi, S. Tognazzi, The Coming Age of Adversarial Social Bot Detection, First Monday, 2021.
[19] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[20] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, DNA-inspired online behavioral modeling and its application to spambot detection, IEEE Intell. Syst. 31 (5) (2016) 58–64.
[21] I. Dimitriadis, K. Georgiou, A. Vakali, Social botomics: A systematic ensemble ML approach for explainable and multi-class bot detection, Appl. Sci. 11 (21) (2021) 9857.
[22] M. Davis, Types of Bots: Categorization of Accounts Using Unsupervised Machine Learning (Ph.D. thesis), Arizona State University, 2019.
[23] X. Ruan, Z. Wu, H. Wang, S. Jajodia, Profiling online social behaviors for compromised account detection, IEEE Trans. Inf. Forensics Secur. 11 (1) (2015) 176–187.
[24] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Emergent properties, models, and laws of behavioral similarities within groups of Twitter users, Comput. Commun. 150 (2020) 47–61.
[25] N. Chavoshi, H. Hamooni, A. Mueen, Debot: Twitter bot detection via warped correlation, in: Icdm, 2016, pp. 817–822.
[26] N. Chavoshi, H. Hamooni, A. Mueen, Identifying correlated bots in twitter, in: International Conference on Social Informatics, Springer, 2016, pp. 14–21.
[27] M. Mazza, S. Cresci, M. Avvenuti, W. Quattrociocchi, M. Tesconi, Rtbust: Exploiting temporal patterns for botnet detection on twitter, in: Proceedings of the 10th ACM Conference on Web Science, 2019, pp. 183–192.
[28] L. Mannocci, S. Cresci, A. Monreale, A. Vakali, M. Tesconi, MulBot: Unsupervised bot detection based on multivariate time series, in: 2022 IEEE International Conference on Big Data (Big Data), IEEE, 2022, pp. 1485–1494.
[29] K. Lee, J. Caverlee, S. Webb, Uncovering social spammers: social honeypots+ machine learning, in: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2010, pp. 435–442.
[30] O. Varol, E. Ferrara, C. Davis, F. Menczer, A. Flammini, Online human-bot interactions: Detection, estimation, and characterization, in: Proceedings of the International AAAI Conference on Web and Social Media, Vol. 11, 2017.
[31] S. Yardi, D. Romero, G. Schoenebeck, et al., Detecting Spam in a Twitter Network, First monday, 2010.
[32] M. Kouvela, I. Dimitriadis, A. Vakali, Bot-Detective: An explainable Twitter bot detection service with crowdsourcing functionalities, in: Proceedings of the 12th International Conference on Management of Digital EcoSystems, 2020, pp. 55–63.
[33] M. Zago, P. Nespoli, D. Papamartzivanos, M.G. Perez, F.G. Marmol, G. Kambourakis, G.M. Perez, Screening out social bots interference: Are there any silver bullets? IEEE Commun. Mag. 57 (8) (2019) 98–104.
[34] C. Shao, G.L. Ciampaglia, O. Varol, K.-C. Yang, A. Flammini, F. Menczer, The spread of low-credibility content by social bots, Nat. Commun. 9 (1) (2018) 1–9.
[35] Z. Chu, S. Gianvecchio, H. Wang, S. Jajodia, Detecting automation of twitter accounts: Are you a human, bot, or cyborg? IEEE Trans. Dependable Secur. Comput. 9 (6) (2012) 811–824.
[36] C.A. Davis, O. Varol, E. Ferrara, A. Flammini, F. Menczer, Botornot: A system to evaluate social bots, in: Proceedings of the 25th International Conference Companion on World Wide Web, 2016, pp. 273–274.
[37] K.-C. Yang, O. Varol, P.-M. Hui, F. Menczer, Scalable and generalizable social bot detection through data selection, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 34, 2020, pp. 1096–1103.
[38] K.-C. Yang, O. Varol, C.A. Davis, E. Ferrara, A. Flammini, F. Menczer, Arming the public with artificial intelligence to counter social bots, Hum. Behav. Emerg. Technol. 1 (1) (2019) 48–61.

[39] J. Echeverrïf!' a, E. De Cristofaro, N. Kourtellis, I. Leontiadis, G. Stringhini, S. Zhou, LOBO: Evaluation of generalization deficiencies in Twitter bot classifiers, in: Proceedings of the 34th Annual Computer Security Applications Conference, 2018, pp. 137–146.
[40] C. Hays, Z. Schutzman, M. Raghavan, E. Walk, P. Zimmer, Simplistic collection and labeling practices limit the utility of benchmark datasets for Twitter bot detection, in: Proceedings of the ACM Web Conference 2023, 2023, pp. 3660–3669.
[41] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial machine learning at scale, 2016, arXiv preprint arXiv:1611.01236.
[42] B. Wu, L. Liu, Z. Dai, X. Wang, K. Zheng, Detecting malicious social robots with generative adversarial networks, KSII Trans. Internet Inf. Syst. (TIIS) 13 (11) (2019) 5594–5615.
[43] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, arXiv preprint arXiv:1411.1784.
[44] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Social fingerprinting: detection of spambot groups through DNA-inspired behavioral modeling, IEEE Trans. Dependable Secure Comput. 15 (4) (2017) 561–576.
[45] Z. Miller, B. Dickinson, W. Deitrick, W. Hu, A.H. Wang, Twitter spammer detection using data stream clustering, Inform. Sci. 260 (2014) 64–73.
[46] S.T. Jan, Q. Hao, T. Hu, J. Pu, S. Oswal, G. Wang, B. Viswanath, Throwing darts in the dark? detecting bots with limited data using neural data augmentation, in: 2020 IEEE Symposium on Security and Privacy (SP), IEEE, 2020, pp. 1190–1206.
[47] C. Yin, Y. Zhu, S. Liu, J. Fei, H. Zhang, An enhancing framework for botnet detection using generative adversarial networks, in: 2018 International Conference on Artificial Intelligence and Big Data (ICAIBD), IEEE, 2018, pp. 228–234.
[48] J. Ma, W. Gao, K.-F. Wong, Detect rumors on twitter by promoting information campaigns with generative adversarial learning, in: The World Wide Web Conference, 2019, pp. 3049–3055.
[49] K.P. F.R.S., LIII. On lines and planes of closest fit to systems of points in space, Lond., Edinb., Dublin Philo. Mag. J. Sci. 2 (11) (1901) 559–572, http://dx.doi.org/10.1080/14786440109462720.
[50] A. Antoniou, A. Storkey, H. Edwards, Data augmentation generative adversarial networks, 2017, arXiv preprint arXiv:1711.04340.
[51] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, in: International Conference on Machine Learning, PMLR, 2017, pp. 2642–2651.
[52] S. Cresci, R. Di Pietro, M. Petrocchi, A. Spognardi, M. Tesconi, Fame for sale: Efficient detection of fake Twitter followers, Decis. Support Syst. 80 (2015) 56–71.
[53] P.-C. Lin, P.-M. Huang, A study of effective features for detecting long-surviving Twitter spam accounts, in: 2013 15th International Conference on Advanced Communications Technology (ICACT), IEEE, 2013, pp. 841–846.
[54] M. Mccord, M. Chuah, Spam detection on twitter using traditional classifiers, in: International Conference on Autonomic and Trusted Computing, Springer, 2011, pp. 175–186.
[55] C. Yang, R. Harkreader, G. Gu, Empirical evaluation and new design for fighting evolving twitter spammers, IEEE Trans. Inf. Forensics Secur. 8 (8) (2013) 1280–1293.
[56] H. He, Y. Bai, E.A. Garcia, S. Li, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, in: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.
[57] G.E. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newsl. 6 (1) (2004) 20–29.
[58] N. Patki, R. Wedge, K. Veeramachaneni, The synthetic data vault, in: 2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA), 2016, pp. 399–410, http://dx.doi.org/10.1109/DSAA.2016.49.
[59] F.J. Massey Jr., The Kolmogorov-Smirnov test for goodness of fit, J. Am. Stat. Assoc. 46 (253) (1951) 68–78.
[60] J.M. Joyce, Kullback-leibler divergence, in: International Encyclopedia of Statistical Science, Springer, 2011, pp. 720–722.
[61] K. Lee, B.D. Eoff, J. Caverlee, Seven months with the devils: A long-term study of content polluters on twitter, in: Fifth International AAAI Conference on Weblogs and Social Media, 2011.
[62] J. Diesner, E. Ferrari, G. Xu, Proceedings of the 2017 IEEE/ACM international conference on advances in social networks analysis and mining 2017, 2017.
[63] S. Cresci, F. Lillo, D. Regoli, S. Tardelli, M. Tesconi, FAKE: Evidence of spam and bot activity in stock microblogs on Twitter, in: Twelfth International AAAI Conference on Web and Social Media, 2018.
[64] A. Shoshan, N. Bhonker, I. Kviatkovsky, G. Medioni, Gan-control: Explicitly controllable gans, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 14083–14093.

Ilias Dimitriadis is a Post-doctoral researcher at the Department of Informatics, Aristotle University, Greece. He holds a Ph.D. from the Informatics Department, Aristotle University of Thessaloniki, an M.Sc. degree in Information Communication Technologies (International Hellenic University) and a B.Sc. in Physics (Aristotle University). His current research interests include large graph mining and analytics, big data mining, social network, bot detection in Social Networks, data analytics and visualization.

George Dialektakis is a researcher at the Department of Informatics, Aristotle University. He holds an M.Sc. degree in Data and Web Science (Aristotle University) and a B.Sc. in Electrical and Computer Engineering (Technical University of Crete). His current research interests include deep learning, adversarial machine learning and data analytics.

Athena Vakali is a Professor at the Department of Informatics, Aristotle University, Greece, where she leads the DataLab research group. She holds a Ph.D. degree in Informatics (Aristotle University), an M.Sc. degree in Computer Science (Purdue University, USA), and a B.Sc. in Mathematics (Aristotle University). Her current research interests include big data mining and analytics, Future Internet applications and enablers, online social networks mining, as well as on online sources data management on the cloud.