applied sciences

MDPI

Article

Social Botomics: A Systematic Ensemble ML Approach for Explainable and Multi-Class Bot Detection

Ilias Dimitriadis *, Konstantinos Georgiou and Athena Vakali

Department of Informatics, Aristotle University of Thessaloniki, 54124 Thessaloniki, Greece; konsgeor@csd.auth.gr (K.G.); avakali@csd.auth.gr (A.V.)
* Correspondence: idimitriad@csd.auth.gr

Abstract: OSN platforms are under attack by intruders born and raised within their own ecosystems. These attacks have multiple scopes from mild critiques to violent offences targeting individual or community rights and opinions. Negative publicity on microblogging platforms, such as Twitter, is due to the infamous Twitter bots which highly impact posts' circulation and virality. A wide and ongoing research effort has been devoted to develop appropriate countermeasures against emerging "armies of bots". However, the battle against bots is still intense and unfortunately, it seems to lean on the bot-side. Since, in an effort to win any war, it is critical to know your enemy, this work aims to demystify, reveal, and widen inherent characteristics of Twitter bots such that multiple types of bots are recognized and spotted early. More specifically in this work we: (i) extensively analyze the importance and the type of data and features used to generate ML models for bot classification, (ii) address the open problem of multi-class bot detection, identifying new types of bots, and share two new datasets towards this objective, (iii) provide new individual ML models for binary and multi-class bot classification and (iv) utilize explainable methods and provide comprehensive visualizations to clearly demonstrate interpretable results. Finally, we utilize all of the above in an effort to improve the so called Bot-Detective online service. Our experiments demonstrate high accuracy, explainability and scalability, comparable with the state of the art, despite multi-class classification challenges.

Keywords: anomaly detection; bot detection; data mining; explainable AI; social network analytics; supervised classification; Twitter

Citation: Dimitriadis, I.; Georgiou, K.; Vakali, A. Social Botomics: A Systematic Ensemble ML Approach for Explainable and Multi-Class Bot Detection. *Appl. Sci.* **2021**, *11*, 9857. https://doi.org/10.3390/app11219857

Academic Editors: Federico Divina and Gianluca Lax

Received: 28 July 2021
Accepted: 13 October 2021
Published: 21 October 2021

1. Introduction

OSNs have gone well beyond information circulation, becoming a vital part of all human activities. The power of OSNs in diverting public opinion can be compared to and is frequently even greater than the power of traditional mass media or other forms of social interaction [1]. Users of varying intentions and origins continuously interact and impact public opinion trends, sometimes with disruptive consequences. The public openness of OSN platforms has given ground to the rise of automated accounts which mimic human behavior. Such accounts, also known as *social bots*, are machine or human controlled software, either benevolent or malevolent, depending on their intentions [2]. The impact of such accounts has attracted the attention of the scientific community, as their spread and reach is constantly increased, while their activities are continuously refined and transformed.

Background. Particularly in Twitter, where the target audience is more vocal on differing opinions and ideologies, bots find fertile ground and ample opportunities in planting discord, stealing sensitive data and attracting users for personal gain. In many cases, such as the recent US elections, official government authorities are becoming increasingly aware of the interference of social bots in the voting process, by influencing the opinions of individuals [3]. As OSN users become more engaged and absorbed in using media, different social bots with various intentions, focused on specific purposes, emerge. For example,

there may be bot accounts, the primary purpose of which is to continuously spam explicit, questionable or misleading content in an attempt to spread false rumors and news, known as *spam bots* [4–7]. Another emerging instance of accounts are politically infused bots, namely *political bots*, that get actively involved in elections and political debates in order to sow discord and deconstruct democratic procedures [8–12]. There are also bot-assisted humans or human-assisted bots, known as *cyborgs* [13] and even *self-declared* bots[9], which are accounts that announce their bot identity.

It is evident that the threat that bots impose on OSNs is alarming as they frequently create practical confrontations between individuals. These threats expand beyond individual harm and can create rifts and divisions between communities, polarizing the public [14,15]. Social bots have frequently been employed in voicing negative opinions on sensitive matters (e.g., climate change) [16], causing harmful consequences. Moreover, they are also a challenge for the global economy, as their intrusion in internet traffic [17] and online transactions jeopardize the transparency of economic disputes and the protection of global economic data [18,19]. Characterizing all social bot accounts as ill-willed would be inaccurate, as there are many that are neutral or even beneficial for users [20]. However, the unfortunate reality is that the majority of social bot accounts do not have benign purposes [21].

Challenges. Recent research has shown that in order to avoid being suspended, social bot accounts can potentially learn to adapt in human behaviors; they show evidence of evolution [22] and linger in social media for prolonged periods of time, further expanding their functions. What is also worrying, is the fact that humans seem to have a hard time distinguishing bot from human accounts [23]. Although Twitter itself has put a lot of effort into detecting and removing fake and bot accounts [24–26], the issue still remains. Thus, detecting and tackling bot accounts is of vital importance for the structure of society to function properly. In accordance with this purpose, existing and recent bibliographies have focused mainly on developing robust frameworks and algorithms that can distinguish human from bot accounts and pinpoint some guidelines for future detection and suspension of such accounts. Dominant among the proposed methodologies are mainly supervised and some unsupervised ML approaches which combine network dynamics for a well-rounded framework.

While significant research has been undertaken in detecting social bot accounts, there is a notable gap in distinguishing different types of bots and inferring what differentiates them [22]. The challenges that arise in this type of detection is that the existing developed models are suited for the binary classification of accounts and cannot be easily adjusted for the refined needs of detecting different type of bots. To facilitate the identification of different bots and eventually dictate new guidelines for tackling these accounts, a new ML approach is highly important. *This work is motivated by the strong need of a refined framework* which will advance ML algorithms to surpass the limitations of a simple bot or human distinction.

Another potential challenge is that behavioral information for the different characteristics of bot types is limited, and thus determining their varied traits requires more than a simple ML algorithm. Currently, there is restrained support in explaining the attributes that differentiate bot accounts (e.g., a spam bot from a simple bot). This creates a cloud of confusion in present bot detection models, as the plain labeling of bot or human cannot reveal the category of a bot account, nor the characteristics that led to this categorization. The present bibliography on bot detection adduces an impressive number of features employed in the detection of bots and humans. These features belong in several categories and serve different purposes on predicting the probability of an account being a bot. However, an open subject of research is the examination of the importance of these features in prediction and the deployment of an inferential process that could determine which features truly matter during a bot type classification. This work builds an efficient bot type detection approach which is more robust, exploiting the emerging field of explainable ML to provide feedback on the different classification of bot types.

Contributions. The above interconnected challenges have motivated this work which aims to cover the current research gap of multi-class bot detection, while also enhancing and improving current state-of-the-art practices with explainable and feature engineering methods. In summary, the major contributions of this work are the following:

1. Newly introduced bot categories and extensive exploratory analysis of datasets: We perform an exploratory analysis of all open datasets and reveal that most of them are outdated and do not keep pace with the evolutionary nature of bots (as discussed in Section 3). Based on this analysis, we define new bot categories that have not been detected in previous bibliographies and map them to the training datasets of our classifiers, enriching a robust multi-class detection schema.
2. New ML models: We implement a set of classifiers combined with imbalance handling techniques, which in contrast to most previous works, are able to identify different types of bots. Our approach competes with existing state-of-the-art ML approaches, despite the multi-class classification challenges, as presented in Section 5.2. We also provide a publicly available web application and open API which uses these models for inferring both the probability of an account being a bot and the probability of it being a specific type of bot, at bot-detectiveV2.csd.auth.gr (accessed on 1 October 2021). We open our source code at https://github.com/idimitriadis/Botomics (accessed on 1 October 2021).
3. Two-stage multi-feature engineering: A multi-feature engineering process is introduced, aiming to examine the performance of the classifiers under different combinations of feature categories (as described in Section 4.1). Our experimentation has shown that even with few features, we achieve a high classification performance.
4. An explainable and exploratory ML framework: In contrast to most previous work that had not emphasized the explainability of their results, we rationalize the predictions of the proposed classifiers (see Section 6). To that end, we utilize popular explainability frameworks, in conjunction with statistical methods, in order to profile each bot category, based on the features that mostly affect their prediction. Apart from providing bot probability scores with the use of the API, we also offer a web portal which also returns textual feature explanatory snippets to unfold the classification prediction steps.
5. Reproducible data sharing: We build and open two new datasets of accounts enhanced with the newly detected bot categories, which have not been included in the current status of bot and human account datasets. These datasets augment the different sources of bot types, contributing to further scientific research in the field.

The remainder of this paper is organized as follows: Section 2 summarizes the related work. Section 3 discusses the process of data collection and the results of our exploratory analysis, while Section 4 outlines our feature engineering approach and ML methodology. Section 5 presents the experimental results of our research and Section 6 describes our explainability approach. Finally, Sections 7 and 8 include the discussion, conclusions and future potential regarding this paper.

2. Related Work

Before delving into the developed methods and algorithms for bot detection, it is important to highlight the nature of a bot and its functionalities. According to Botwiki (https://botwiki.org/bots/, accessed on 1 October 2021), a bot is "a software application that runs automated tasks over the Internet. Typically, bots perform tasks that are both simple and structurally repetitive, at a much higher and intense rate than would be possible for a human alone." A different interpretation is that a bot is "a software application that is programmed to do certain tasks" (https://www.cloudflare.com/learning/bots/what-is-a-bot/, accessed on 1 October 2021). Essentially, bots are automated tools that function without the need for human intervention or control, except when control is defined by research objectives [13]. Their purpose is to perform repetitive tasks that would seem mundane or time consuming for humans, exhibiting much faster rates than a human

account. A practical and more humanly intuitive notion is that bots are "predictable automatons that do not have the capacity for emotions, meaning-making, creativity, and sociality" [27].

Bots exponentially gain ground in OSNs and their dominance is already evident. For example, according to a recent survey, two thirds of the URLs shared in Twitter feeds are posted by bots [28]. As the popularity of bots grows and social media platforms routinely benefit or are harmed by their use, researchers have turned their attention to tracking bots, especially on Twitter. During the past years, there have been various attempts at detecting a bot, or specific categories of bots. In general, the timeline of bot detection spans a decade, with research studies increasing in later years [22]. As the popularity of bots grows and social media platforms routinely benefit or are harmed by their use, researchers have turned their attention to tracking bots, especially on Twitter. This can be further highlighted by the wide influence of bots in presidential elections [29,30], financial markets [19], pandemic related news [31] and cryptocurrency manipulation [32] as well as the various and different bot categories [2], and particularly social bots [33].

Recent studies have highlighted that the majority of existing literature relies either on ML approaches or unsupervised methodologies in order to construct robust implementations that fulfill this purpose. There have also been attempts that provide web-based tools which gather data related to user accounts on social media from specialized APIs and infer if the account is a bot. In the following we will review the latest and most important bot detection approaches.

Regarding supervised approaches and spam bots, Ref. [34] is the first attempted research work that refers to spam bot detection via an ML Classifier. Apart from classifying Twitter accounts as spam bot or human, this work takes the first step towards exploring the major characteristics of spam bot accounts, such as tweet frequency, longevity and networks of followers. Spammers and social bots are also the central point of another paper [35], where the authors construct fake accounts to attract spammer activity and exploit the accumulated data to construct robust classifiers based on textual and network features. However, it is fairly evident that, as detection methodologies evolve, so do bot accounts, which adapt and develop evasion strategies. Thus, Ref. [36] proposes new features that are not affected by evasion strategies and are either graph related (clustering coefficient, betweenness centrality) or neighbor related (followers of neighbors, tweets of neighbors, etc.). This meticulous construction of features eventually yields good results, with low false positive rates. Several studies [37–39] use Random Forest classifiers and rely on well-established user and content-based features for detection, while also examining the robustness of the utilized features. In addition, another approach [40] enhances the current frameworks by adding behavioral features, in an effort to move away from single account detection and instead track groups of spambots. Finally, Ref. [41] proposes a hybrid approach that primarily takes into account the interactions of a bot account with his followers and combines them with predefined features, constructing classifiers with high accuracy levels.

As far as social bots are concerned, a notable effort by [10] constructs various novel datasets that contain social bots which act as fake follower inflators for Twitter accounts. Along with this addition, they simultaneously examine the efficiency of different feature sets in conjunction with several different classifiers. The Random Forest classifier is once more proven dominant. BotOrNot [5,8,9,42], a more refined framework that utilizes over 1000 features distributed in several categories (user, content, network, sentiment) is also presented as a state-of-the-art solution in bot detection. In addition, BotorNot is one of the few solutions that actually perform multi-class classification of bots. The authors of [28] propose a full-fledged framework that utilizes 1150 features and gives emphasis on the interactions between a genuine human account and a social bot account. More recent efforts [43–45], move away from a simple classifier detection and instead focus on the malicious effect that social bots have on Twitter, by analyzing political and marketing campaigns or pointing out the biggest challenges in their detection, such as their evolving

behavior, which largely imitates human actions. In addition, the latest methodologies include subsetting the bot accounts in order to select the best dataset setup for improved classifier accuracy [8] and adversarial algorithms [46,47] that synthetically create social bot accounts to directly interact with other accounts, learn their strategies and improve detection rates. Apart from works that focus solely on bots, Ref. [13] introduces the distinction between humans, bots and cyborgs (human controlled bots) and proposes an entropy-based framework to classify them.

Unsupervised approaches are still gaining ground, thus research ventures are notably limited in comparison with supervised methodologies. Researchers in [48] use statistical inference and behavioral features to trace clusters of extroverted users, labelling them as social bot accounts. Behavioral similarities are also the primary subject of [49] which via unsupervised simulations reveal suspicious activities. An online framework, namely DeBot, that does not require labeled data and is focused on correlated activities between accounts was presented in [50]. The rationale of this approach is that two accounts that appear to have highly similar activities are likely to be bots. The metric of temporal synchronicity is the core aspect of Debot in detecting simultaneous changes in the state of discrete elements. In the same spirit Rtbust [51], is another unsupervised framework that exploits temporal patterns (retweets, mentions, likes) to highlight malicious activity.

Recent studies have explored group approaches, that cannot be considered either supervised or unsupervised. Their focus is drawn towards botnets, which can be conceptualized as groups of bots that act together to achieve a common purpose. Notable in these methodologies are network approaches that attempt to detect synchronized and suspicious account behavior [52–54], analyze the connectivity of such accounts and propose appropriate countermeasures for their detection and deactivation.

Despite their robustness and effectiveness, bot detection methodologies have recently been criticized for some crucial deficiencies. As bot behaviors evolve over time, the produced methodologies are sometimes unable to stay up to date and are rendered useless in detection campaigns [6], while it has been pointed out that focusing on the evasion strategies of bots rather than simply detecting their presence is the next logical step towards the evolution of the field [55]. In addition, the need for training the classifiers on previously unknown bot classes in order to potentially increase their efficiency is raised as an urgent solution to restrained accuracy scores [56]. One of the main issues that emerge is the lack of credible, annotated datasets, on which supervised solutions are trained [57]. At this point it should also be mentioned that only a few works pay attention to the intrepretability of their results [8,58,59], highlighting the need for more explainable models. Last but not least, bot detection platforms have been proven to be prone to increased false positive rates [60], a fact that hinders and reduces their credibility.

Our work provides a holistic approach in an attempt to fill the identified gaps, by broadening the scope of its research, rather than focusing on solving a particular issue. More specifically, our work initially presents an exploratory analysis of the open datasets, which have been used by all the previous works, and uncovers that most of these data are unavailable or outdated. At the same time, as recommended above, we propose a new validated division of different bot types, based on the description of each available dataset. Although feature engineering has been extensively discussed, we extend this work by providing a complete analysis for features used in binary- and multi-class bot classification tasks, while we also offer credible, interpretable results. Finally, we present a set of both binary- and multi-class ML models, inspired by [9], which offer competing results, in comparison to current state-of-the-art approaches.

3. Current State of Bot-Related Data

In this section, we present the first two blocks of the proposed research methodology modular framework, as presented in Figure 1.

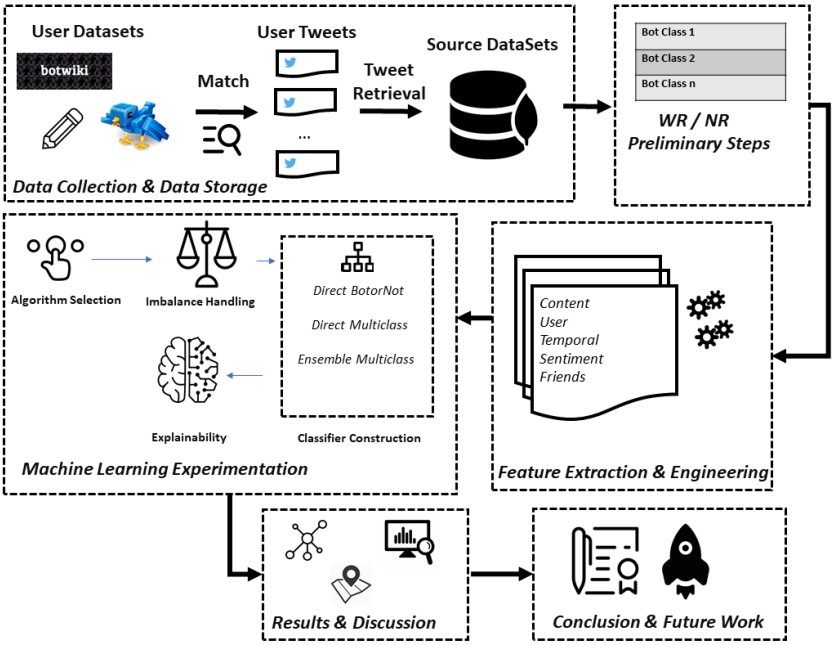

Figure 1. Outline of the proposed framework architecture. The data collection and analysis phase is discussed in Section 3, while the ML and feature engineering approach is further discussed in Section 4.

In Section 3.1, we describe the data collection process and the results of our exploratory analysis. The analysis of the datasets allows us to proceed with a further categorization of bot types, as presented in Section 3.2.

3.1. Data Collection and Exploratory Analysis

As already mentioned, the volume of supervised ML methodologies focused on bot detection is constantly growing [22], and most approaches use already available labeled datasets. Most of these datasets can be found in the Data Repository section of Botometer [5,8,42]. This repository contains Twitter accounts and their identification numbers (Twitter user IDs), while also characterizing them as "human" or "bot". Apart from collecting data from Botometer, we also included two new datasets that were the result of a manual account search in Twitter. We opted to follow this approach to detect new types of accounts that were not contained in the Botometer repository, namely news agencies and companies that were labeled as "cyborgs", which also include celebrity accounts. This, in turn, allowed us to expand the defined bot categories that will be explained in subsequent sections. In Table 1, we present an overview of the characteristics of each dataset.

Having accumulated the targeted accounts and their IDs, instead of simply scraping Twitter for content (an action that is prohibited by the platform anyway), we exploited existing services provided by Twitter that enable rapid and easy collection of tweets for a given account. We "hydrated" each account with its most recent 1000 tweets (other approaches like [42], use a few hundred tweets), unless it had fewer tweets in the entire timeframe of its existence. Naturally, as Twitter has launched a quite ambitious campaign to counter bot activity, many accounts had been suspended and were thus nonfunctional. As presented in Table 1, although there is a plethora of labeled Twitter users in the open datasets, most of these accounts have either been deleted or suspended.

Table 1. Outline of collected datasets. Columns **H**, **B** and **C** refer to Humans, Bots and Cyborgs, respectively. The actual data are those presented as **Av**. Only **40%** of the total shared labeled accounts are still available in Twitter.

#	Dataset Name	H/Av.	B/Av.	C/Av.	Description
1	This paper (2021)	-	-	232	Popular companies
2	This paper (2021)	-	-	160	News agencies
3	Sayyadiharikandeh et al. [9]	-	585/442	-	Active political bots
4	Yang et al. [8]	-	1488/1198	-	Self-declared bots Botwiki
5		2000/1976	-	-	Verified human accounts
6		8092/7369	42,466/42	-	Political accounts
7	Yang et al. [5]	386/331	143/101	-	Labeled by Botometer
8		-	21,964/1733	-	Porn spammers
9		-	62/13	-	Active political bots
10		-	1088/707	-	Vendor-Purchased bots
11		-	-	5971/5499	Celebrity accounts
12	Mazza et al. [51]	368/313	391/311	-	Bot and human accounts
13	Cresci et al. [18]	7479/5797	18,508/6752	-	Stock related
14	Varol et al. [28]	1747/1291	826/657	-	Simple bots and humans
15	Cresci et al. [6,61]	-	1506/1356	-	Bots spamming content
16		3474/2397	-	-	Genuine Twitter accounts
17		-	5915/4166	-	Bots "hunting" followers
18	Gilani et al. [62]	1522/1281	1130/980	-	Simple bots and humans
19	Cresci et al. [10]	1481/1162	-	-	Italian elections E13
20		469/417	-	-	Fake Project TFP
21		-	1337/60	-	InterTwitter INT
22		-	845/588	-	Technology spam TWT
23		-	1170/28	-	Fast Followerz FSF
24	Lee et al. [4]	19,277/8238	22,224/13,394	-	Content polluters and humans
#	Total Accounts	46,295/30,572	121,648/32,528	6363/5891	174,306/68,991

More precisely, out of more than 170 K of accounts that have been labeled in previous studies, only 69K of them are still available in Twitter. In an effort to help other researchers, we share the IDs of the accounts that are still valid at https://github.com/idimitriadis/Botomics (accessed on 1 October 2021). Additionally, we highlight that although these remaining accounts are still available, this doesn't mean that they are still active in posting content or interacting with other users. For this purpose, we continued with a further exploratory analysis of each dataset. For every account, we extracted the date of their latest tweet, to discover the last time they posted some type of content online. Figure 2, indicates the total number of accounts, the number of accounts per type and the year they posted their latest tweet.

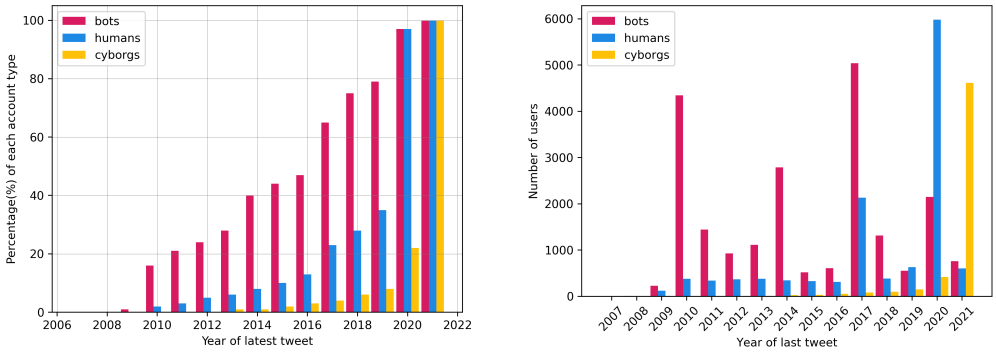

Figure 2. The figure on the left shows the percentage of accounts that stopped posting content per year. Interestingly more than 75% of bots remained inactive after 2017. The figure on the right shows the number of each type of account to the year they posted their last tweet. We observe that most labeled bot accounts that are included in the public datasets are no longer active.

It is obvious, that most bot accounts are inactive, while humans and cyborgs continue their online activity. This fact, points out a problem that has been discussed a lot by other researchers during the last years. Thus, the availability and credibility of up-to-date data has become a major issue and since the evolving nature of bots renders this data out of date, there is a strong need to constantly update and open bot-relevant databases.

There have been various efforts from researchers to provide up-to-date labeled data, such as DeBot [50] which supposedly offers an API and access to lists of users that have been deleted by Twitter after their detection, or Botometer [5,8,42] which provides a public API which evaluates the possibility of an account being a bot. Unfortunately, such services are either no longer functional or highly criticized, thus they do not offer solid ground to build on [6,55]. Here, we will only use publicly available shared data that have been used by other researchers, and which have been filtered in such a way that only include accounts that have not been either suspended or deleted. Unlike most other previous work, which use data that are no longer available, we follow a more realistic approach, based on the truly and actually available data.

3.2. Not All Bots Are Created Equal

Before proceeding into the feature extraction process, we defined multiple classes for the retrieved accounts. Thus, we proceeded beyond a simple binary categorization of either bot or human to explore different bot classes which are presented in Table 2, along with a short description and the datasets that comprise them. The numbers of the datasets, refer to the serial number of each dataset, as presented in Table 1.

Table 2. Mapping of bot types and datasets and their description. The numbers of the datasets, refer to the serial number of each dataset, as presented in Table 1.

Bot Class	Description	Class Datasets (#)	Instances
Spam Bot	Accounts that post spam content	24,22,15,8	17,071
Social Bot	Bots that try to attract followers	17,13,10,23	11,653
Political Bot	Bots that deal with politics	3,6,9	497
Cyborg	Human monitored bots	1,2,11	5891
Self-declared	Accounts that state they are bots	4	1198
Other Bot	Other type of simple bots	7,12,18,21,14	2109
Human	Genuine human accounts	5,6,7,12,13,14,16,18,19,20,24	30,752

As presented in Section 1 and Table 2, the following bot classes appear in recent literature and have been retrieved in combination with the information contained in the datasets that we utilized.

- Spam bots encapsulate every type of bot that is related to continuously posting spam content.
- Social bots are related to impersonators, influence bots and paybots.
- Political bots are a rather unique class, including accounts that have been used for political purposes.
- Self-declared bots refer to accounts that identify themselves as bots.
- Cyborgs include celebrities, news agencies and organizations, and as discussed in Section 2, they have not been included in recent multi-class bot detection approaches.
- Other bots include all bots that do not fall into any of the previous categories according to the dataset description.

It is important to clarify that across these umbrella classes, there are malevolent and benevolent bots as well.

To measure the efficiency of the defined bot classes we trained six binary Random Forest classifiers, one for each bot type. As depicted in Section 2, Random Forests have proven to be quite effective for similar bot classification tasks. The hyperparameters of each classifier are similar to the ones defined in Section 5, while the features that have been used for training our classifiers are thoroughly presented in Section 4.1. Each classifier has been trained on a balanced subset of data, using ADASYN [63] as discussed in Section 4.2. Each subset includes human accounts and instances of each bot type, split into train and test samples. The produced classifiers were thus trained on each subset and tested across all the others. The rationale behind this approach was to evaluate the cross-type performance of the individual binary classifiers, validating our hypothesis.

We measured the performance of each bot classifier targeting the same or other bot types by calculating the precision and recall metrics. We can easily observe from our exploratory analysis (Figure 3) that classification precision and recall scores are strong for every classifier that targets the same bot type as the one it has been trained on. This is true for all bot types, as indicated by the dark colored diagonal blocks in Figure 3, except for "other bots".

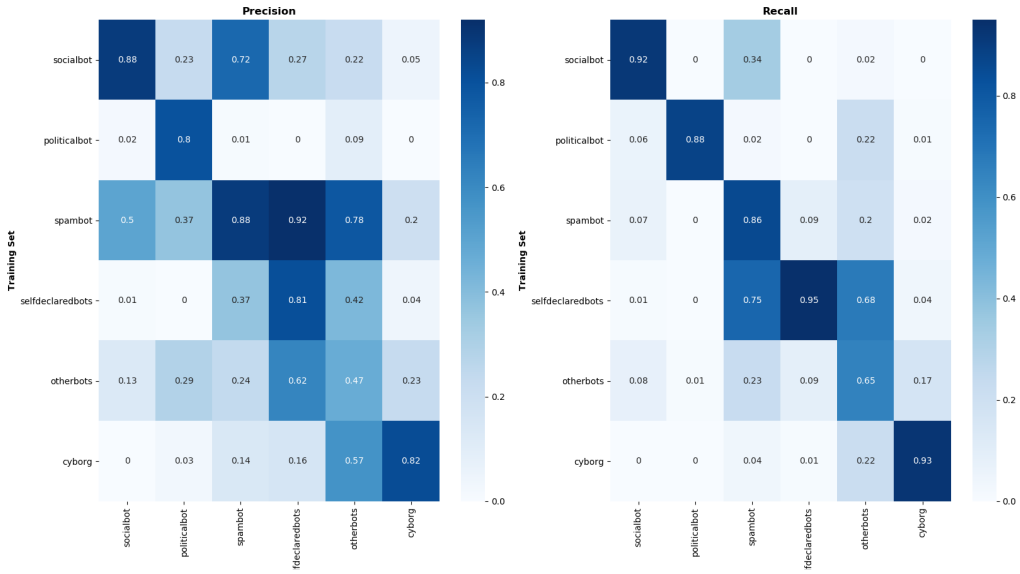

Figure 3. Precision and Recall of binary human or i bot type classifiers, trained on the D_i dataset and tested on other bot types. The low cross-type performance showcases the need of defining different bot types.

On the contrary, cross-type performance is really low, which highlights the different behavior of bots. It should also be noted, that the only datasets where other classifiers' performance somehow increases, is the one for the "other bots" and the one for "self-declared bots". In summary, our main observations are the following:

- Low cross-type performance designates the need for the distinction of bots into separate types.
- Other bots: taking into account that this bot type label had been assigned to datasets that did not include much information, so as to assign them to one of the other categories, we can easily work out that this class contains samples of the other bot types as well.
- The high precision observed in the self-declared dataset when tested with other classifiers is due to the fact that it includes bots with activites quite similar to those of spam bots and that the other bots' datasets most probably include self-declared bots as well.
- The majority of labels that we manually assigned, correctly correspond to different kinds of bots, pointing out the lack of generalization of each bot-type-specific model.

In the following section, we discuss the various feature types, the feature extraction process and how these features affect our final dataset.

4. Methodology

In this section, we thoroughly present the feature extraction and feature selection methodology, see Section 4.1, to cover the proper training of our ML models which are described next in Sections 4.2 and 4.3.

4.1. Feature Engineering: A Systematic Approach

In each typical ML task, the selection of appropriate features plays a very important role for building an efficient ML model. In this work, we utilized features that have already been introduced and defined by previous researchers and we have enriched them with additional ones that, to the best of the authors' knowledge, have not been used before. We have identified more than 400 individual features, which can be extracted leveraging the plethora of information included in each Tweet object (https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/tweet (accessed on 1 October 2021)), which also includes the User (https://developer.twitter.com/en/docs/twitter-api/v1/data-dictionary/object-model/user (accessed on 1 October 2021)) object as well.

We followed a systematic approach to categorize the different features, based on their contextual scope. Our approach introduces a simple taxonomy of feature categories which is further described next:

1. Content Features (C): Text-relevant metrics to capture the source data semantics as expressed in tweets and their relevant properties. For example, these reflect semantic elements found in text (e.g., text size) or numerical elements corresponding to the tweet's content (e.g., frequency distributions of punctuation marks, inter-tweet text similarity, etc.).
2. User Features (U): Extracted directly from the Twitter API User object and which cover several characteristics of the account owner. For example, in this class we include attributes of the user's profile description (e.g., the existence of the word "bot" in the description"), numerical features regarding user activity (e.g., the number of followers, followers and the follower/friends ratio) or Boolean values that provide additional information about the user (e.g., if the user is verified or if a location is provided).
3. Temporal Features (T): Features which are exclusively relevant to the timestamps of tweets and retweets and the elapsed time between tweets and retweets in a given time frame (e.g., tweet posting rate per day, maximum tweets per hour, consecutive hours of online activity, etc.).

4. Social Neighborhood Features (N): Metrics relevant to the accounts that are retweeted by the user. They are particularly useful in profiling the surrounding social circle of a user, the user's network, in order to determine the type of account in question. Examples of such features are the percentage of friends that have profile descriptions or the distribution of their tweeting activity.
5. Sentiment Features (S): Indicators in the range of tweets' sentiment characterization. Sentiment analysis (computation of textual neutral, positive or negative sentiment percentage), extended beyond the presence of emojis as indicators of wide emotional expression (e.g., the ratio of total emojis to the length of the tweet's text).
6. Hashtag correlation features (H): Finally, we introduce a set of features that are generated by the network of used hashtags. Consider an undirected graph G, which has as N nodes, the hashtags used by the user and E edges, being defined from a hashtag co-occurrence matrix. Thus, once hashtags h_1, h_2 are present in the same tweet, it holds that $h_1, h_2 \in N$: $e_1 = (h_1, h_2) \in E$. This graph is valuable to calculate graph interconnection properties, such as the number of triangles or the node to edges ratio.

During the feature extraction process, we identified some major bottlenecks which do not allow us to use the whole set features. These issues are described below:

- Source data thread limitations, due to the minimum number of tweets and retweets that are included in the last 1000 tweets of each user, as described in Section 3.1. More specifically, in order to use the features included in the Social Neighborhood (N) category, we expect to have at least two retweets among the tweets that have been collected for each user; failing that, we cannot proceed to the calculation of statistical metrics for the distribution of features.
- Lack of content or activity information, since in our feature extraction process, we found that there are many accounts with no tweets or no retweets. For example, some bots in the Botwiki dataset do not post original tweets but are limited to posting retweets that include a specific keyword. In this case, extracting sentiment features will not be helpful, due to the absence of original content. Moreover, there are some users that do not retweet any content at all; thus, we are not able to identify their network and consequently the extraction of specific features, as we mentioned earlier, is unfeasible.
- Platform-specific limitations, posed by the continuous changes in the Twitter API. For example, we are no longer able to observe the evolution of some metrics that have been proven, by other researchers, to be quite useful as features. Such an example, is the number of friends the user has in a certain snapshot; even if the snapshot refers to the past, the User object (returned by Twitter) is the one currently effective; thus, we are unable to measure the temporal evolution of this metric. We also have to mention that, although Twitter API allows developers to collect the network of each user (friends–followers), the process of doing so is not time-efficient with respect to the official Twitter API rate limits, and therefore it is not considered to be a good solution, due to the need for real-time results.

Two-stage feature extraction : To cover this limitation, we employ a two-stage feature extraction process. Once for the accounts that fulfill all the requirements, using all the available features, and once more using a pruned version of the identified features in order to include all the available accounts in our datasets, regardless of the number of tweets or retweets they have posted. In Table 3, we present an overview of the two feature sets along with the time needed to extract each one. We estimated the average calculation time by computing the average of the time needed to extract each set of features for every user. In an effort to not only reduce computational times in future work, but also to aid researchers in avoiding using redundant features, in the next sections we utilize some feature engineering packages to detect the most important and informative features and use them as a baseline of further research. Moreover, in the ML models (described in the next section), we experiment with all possible combinations of the proposed feature

categories and explore the best combination that yields the most robust performance, by embarking on both binary- and multi-class bot detection (as discussed in Section 1).

Table 3. Overview of feature characteristics per version. The abbreviations of the first column refer to the features category, see Section 4.1. The social neighborhood feature category (N) is available in the full-features version. The pruned-features version is used for accounts that either have no tweets or no retweets.

Feature Category	Number of Features	Average Calculation Time	Version
U	28	0.73 s	Full and Pruned
T	69	0.19 s	Full
T	29	0.14 s	Pruned
N	62	0.015 s	Full
C	190	5.02 s	Full
C	182	7.2 s	Pruned
S	58	0.42 s	Full and Pruned
H	13	0.003 s	Full and Pruned
Total	full: 420/pruned: 309	full: 6.4/pruned: 8.5	-

4.2. Binary Bot Classification

As discussed in Section 2, since we go beyond the conventional binary bot detection approach, we need to define a basic model which will serve as our Baseline Binary Bot detection classifier, namely BBC. Our initial goal is to select the classification algorithm that will ultimately be chosen for further experiments.To achieve this, we experimented with several classification algorithms, namely Logistic Regression (LR), Linear Discriminant Analysis (LDA), Decision Tree Classifier (CART), Multi Layer Perceptron (MLP), AdaBoost (ADA), and Random Forest (RF). We split our dataset into a 75–25% ratio for training and testing the models. For each of these algorithms we performed a GridSearch process for hyperparameter tuning and better parameter selection. The produced models were tested on the test sample (previously unseen data) and they were evaluated by measuring the accuracy using 10-fold cross validation. Eventually, our analysis proved that Random Forest had a clear dominance compared with the other algorithms. Its performance has also been recognized as the most reliable, as discussed in similar research [5,8,9,42,56], and is well aligned with an explainable approach for classification in bot detection experiments. We have included the results of other classification algorithms in Figure 4.

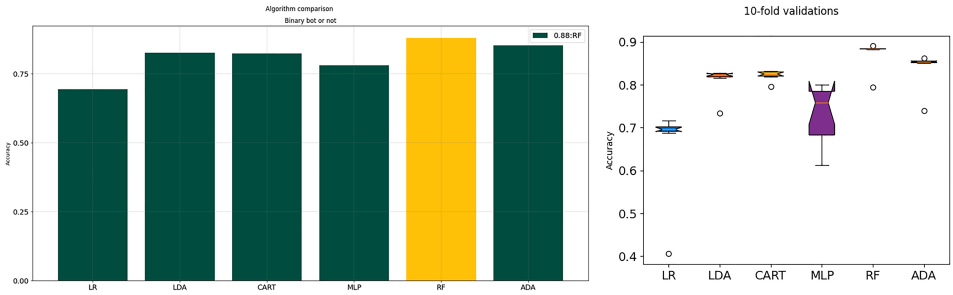

Figure 4. Random Forest performs better than the other classifiers. The figure on the right shows, that apart from highly accurate predictions, Random Forest is among the most stable, with relatively small fluctuations in the 10-fold validation test.

Beyond the bot classification baseline, we followed a proper imbalance handling approach to tune and adjust the uneven amount of instances, as described in Section 3.1. To tackle this problem, right after the selection of the proper classification algorithm,

we experimented with three balancing methods, namely SMOTE-Tomek [64], SMOTE-ENN [65] and ADASYN [63], to select the one that guaranteed a realistic equalizing of cases while maintaining an optimal performance. More specifically, in Table 4, we present the performance metrics for the BBC classifier, using the three different imbalance handling methods that we have already mentioned. Our experimentation runs showed that ADASYN performs better than the other two, preserving a balance between precision and over-fitting.

Next we trained our algorithm on two different datasets, as defined in the feature extraction process. The first dataset includes the users that comply with all the limitations posed by the full-features set, while the second includes users that have at least two tweets or retweets and can be used to extract the shortened feature set. The two final datasets are the following:

1. Full-features Dataset : 46,819 (28,449 humans—18,370 bots)
2. Pruned-features Dataset : 68,437 (30,435 humans—38,002 bots)

Table 4. Performance comparison of the the binary bot classifier, using ADASYN, SMOTE-Tomek and SMOTE-ENN imbalance methods. ADASYN provides better results.

Imbalance Method	Precision	Recall	Accuracy	F1	GMean
ADASYN	**0.895** $\pm$ 0.025	**0.850** $\pm$ 0.03	**0.861** $\pm$ 0.025	**0.870** $\pm$ 0.026	**0.862** $\pm$ 0.025
SMOTE-Tomek	0.894 $\pm$ 0.019	0.837 $\pm$ 0.032	0.855 $\pm$ 0.025	0.864 $\pm$ 0.026	0.857 $\pm$ 0.025
SMOTE-ENN	0.889 $\pm$ 0.02	0.769 $\pm$ 0.018	0.82 $\pm$ 0.019	0.825 $\pm$ 0.018	0.823 $\pm$ 0.019

Finally, with respect to the efficiency of our model, we used different feature combinations, among the ones that were identified earlier. Utilizing feature importance techniques and experimenting with all the possible feature type combinations, we ended up with a smaller set of features which produced similar results in comparison with the full set of features and required much less processing, and thus, calculation time.

4.3. Multi-Class Bot Classification

While the binary classifier was useful in providing an initial estimation, the next step was to develop a multi-class classifier, to separate accounts into different classes defined in Section 3.2. From these classes we excluded the "other bots" class, which, as presented in Section 3.2, seems to be a mixture of all the other bot classes.

In our multi-class classifiers we provided sublabels for the bot class, as defined in Table 2. Similarly to our previous binary bot classification approach, and since the issue of an uneven amount of instances was even more evident in the multi-class classification task, we proceeded with the use of the ADASYN algorithm which performs better than the other algorithms, in this case as well. The ensemble multi-class classifier utilizes both class-specific binary classifiers and multi-class classifiers to predict the probability of an account being a human or a class-specific bot, in combination with the baseline binary bot or human classifier. The rationale behind this approach is to juxtapose the human or bot binary prediction to those that refer to specific bot types. Our ensemble classifier's goal is to initially decide if the account is a human or a bot and subsequently assign a more detailed label to a bot account, as presented in Figure 5.

Next, we present a set of ensemble multi-class classifiers which use the BBC classifier to calculate the probability of an account being a human, namely Human Probability HP, and they operate as described next:

– EMCREST combines the BBC classifier with an one-vs-rest classifier, from which we calculate the probability of an account being a specific type of Bot BP. The one-vs-rest classifier splits the multi-class dataset into multiple binary classification problems and the prediction is made using the model which provides the most prevalent one. The

final prediction of the EMCREST classifier is the most confident prediction between the BP and the HP.

– EMCS uses the BBC and a stacking Random Forest classifier which consists of a set of binary classifiers. At this stage, we implemented this set of binary classifiers for each identified class of bots. Hence, we trained five binary Random Forest classifiers (following the same procedure as in the BBC classifier), each of which is responsible for classifying a user as human or as a specific type of bot. Naturally, we used the appropriate training data for each classifier. For example, the spam bot classifier is trained on data that consists of users which: (a) are labeled as spam bots, (b) are randomly chosen humans, based on the fundamental assumption that humans have a similar online behavior, which differentiates them from bots. The final prediction of the EMCS classifier is the most confident one between the HP and the stacking BP.
– EBBC utilizes the BBC classifier and a set of the class-distinct binary Random Forest, each of which outputs a bot probability BP_i, where i refers to each bot type. The final prediction is the highest probability among the set of $HP, BP_i, ...BP_n$, where n is the number of total bot types (five in total).

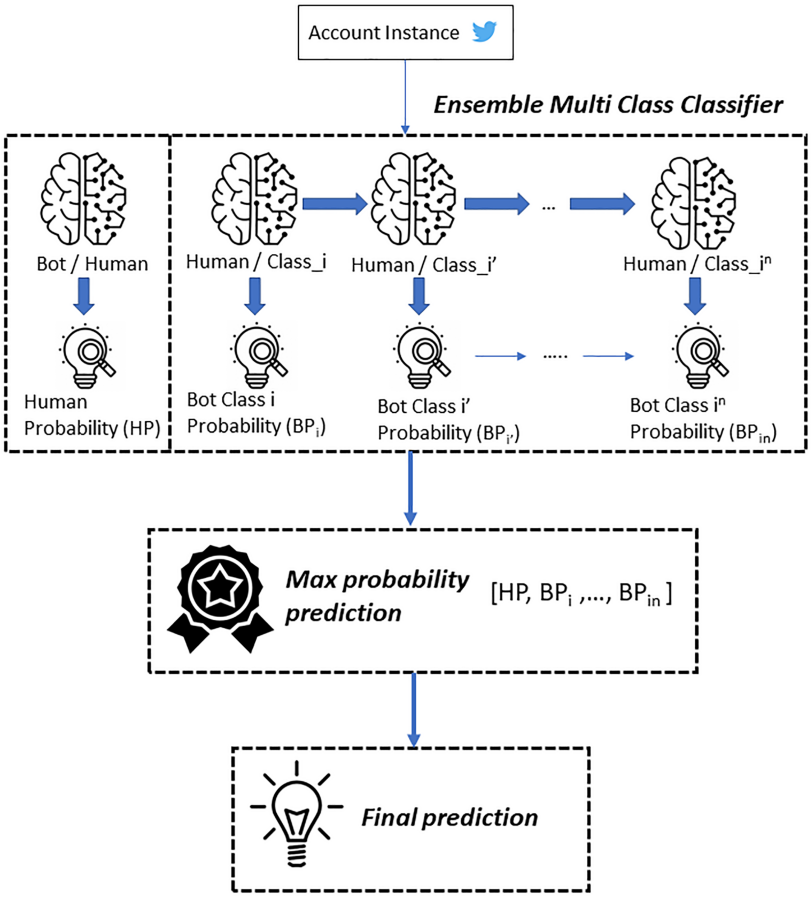

Figure 5. Multi-class Ensemble Execution Pipeline.

The ensemble process does not require any balancing, as this classifier relies on balanced pretrained models. The decision to omit the imbalance handling stage is attributed to the fact that since the constructed ensemble pipeline comprises classifiers that have already been trained on balanced datasets, in the original raw datasets, the model could easily discern the different instances and figure out the class to which each instance belonged. More specifically, as each classifier has been trained to detect a specific class and instance of every class, it would be fairly straightforward for the ensemble classifier to

understand the instance class. More details about the performance of the classifiers can be found in Section 5, where the results and findings of this stage are presented.

5. Experimentation Results

In this section we present the wide range of experiments that have been conducted both at a feature- and model-based level. Apart from providing tuning and performance details on the classifiers that have been used, we also study their behavior based on the combination of the extracted sets of features.

5.1. Baseline Binary Classifier (BBC)

The first ML model built for the task of bot detection was a Random Forest classifier, with 160 decision trees and maximium depth equal to 10, and trained to separate humans from bots, based on the extracted features. As the feature selection experiments take place in subsequent steps, the baseline classifier utilizes all the features from the full-features dataset and the pruned-features dataset. We separated the datasets with a 75–25% ratio for training and testing the model. The training dataset was balanced using the ADASYN algorithm. Arguably, the imbalance of bots and humans was not very restraining, but we chose to balance the dataset in order to avoid a potential bias.

Concurrently with predicting the instances of the test dataset, the most important features were also kept in a separate list. While labels were mapped in the same manner (0 and 1), the number of features was significantly reduced in the model trained on the pruned-features dataset, as discussed earlier. In the following figure, see Figure 6, we present the performance of our binary classifier on the pruned dataset and we show how it changes based on the number of features that we used. We are iteratively removing features, starting from those which have been identified as less important and moving to the most important ones.

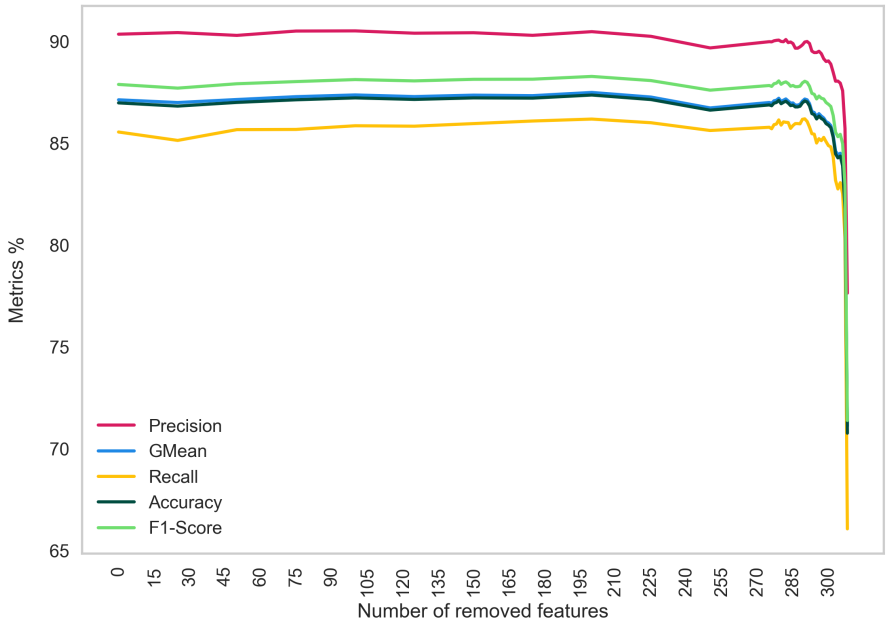

Figure 6. Performance of the BBC classifier for the pruned-features dataset. The x-axis corresponds to the number of features removed in every iteration starting from the least important ones and moving towards the most important. We observe that the performance remains the same, even if we remove more than half of the total features.

As presented in Table 5, we observe that the classifier achieves high precision, G-mean and recall scores. We also discover that the reduction of features does not seriously affect the performance. More specifically, both for the full-features and the pruned-features

dataset we discover that all the evaluation metrics actually remain the same, even for a much smaller set of features, just 145 and 100 out of the total 420 and 309, respectively. The classifier trained on the pruned-features dataset seems to perform better than all the others, using only a small fraction of the total features.

Table 5. Performance of the BBC classifiers in both available datasets.

Data & #Features	Precision	Recall	Accuracy	F1	G-Mean
Full-features [145]	0.787 ± 0.032	0.792 ± 0.054	0.832 ± 0.025	0.789 ± 0.042	0.824 ± 0.035
Pruned-features [100]	**0.893 ± 0.02**	**0.849 ± 0.03**	**0.861 ± 0.025**	**0.87 ± 0.026**	**0.862 ± 0.025**

We also conducted another experiment, to discover which set or which combination of feature types, out of the ones that we defined in Section 4.1, provides better results. For this reason, we trained the BBC classifier using all the possible feature set combinations as input. In Table 6, we demonstrate the results of our experiment. Since we experimented with all the possible combinations, we present only the best ones for each number of combined feature sets. For example, the third column shows the best three feature type combinations per classifier, which is apparently a combination of **U**ser, **T**emporal and **H**ashtag correlation features for the pruned-features dataset. We observe that the accuracy for each feature combination does not fluctuate significantly, regardless of the number of combined feature sets. Moreover, we notice that the User object derived features are present in most cases, followed by those that refer to the temporal activity of the user.

Table 6. Accuracy performance of the BBC classifiers in comparison to the different combinations of feature categories. Each column refers to the best combination for each number of combined feature categories. Apparently, the combination of **U**ser, **T**emporal and **H**ashtag feature categories provides the best results. However, the accuracy does not fluctuate significantly, regardless of the feature category combination that has been used.

Dataset	Best 1	Best 2	Best 3	Best 4	Best 5
Full-Features	U: 0.821	UT: 0.849	UTN: 0.855	UTHN: 0.855	UCTHN: 0.851
Pruned-Features	U: 0.851	UT: 0.87	**UTH: 0.874**	UCTH: 0.87	UCTSH: 0.866

In the next section, following a similar approach, we proceed with the experimentation related to the multi-class bot classifiers.

5.2. Ensemble Multi-Class Classifiers

As presented earlier, we have built three ensemble classifiers (EMCREST, EMCS, EBBC) which combine the BBC classifier with three sets of different classifiers. We performed our experiments using the pruned dataset that was previously described and we used the pruned-feature set, which in the binary bot classification problem seemed to perform better than the full set of features. Following the same procedure as before, we split our data into train and test samples in a 75–25% ratio, and handled the imbalance of our training set using the ADASYN algorithm.

Initially we trained the one-vs-rest algorithm, which actually creates six binary classification problems, where each class "competes" against all the others. The next step involved the training of a stacking Random Forest classifier which consists of five binary Random Forest classifiers, where the competing classes are always "humans" and each other type of bot. These binary classifiers were actually introduced in Section 3.2, where we used them to validate the labels we manually assigned. Having said that, we proceed with the results of our experiments, which are thoroughly presented in Table 7.

Table 7. Performance of the Multi-class classifiers. The EBBC classifier seems to perform much better than all the others.

Classifier	Precision	Recall	Accuracy	F1
one-vs-Rest	0.808	0.888	0.859	0.842
stacking RF	0.855	0.839	0.878	0.864
EMCREST	0.87	0.864	0.881	0.866
EMCS	0.875	0.832	0.883	0.851
EBBC	**0.891**	**0.918**	**0.898**	**0.904**

As we can see, the EBBC classifier's performance is superior to that of all the other classifiers. Evidently, the EBBC classifier would be our choice even if we would just like to distinguish humans from bots in general, since it outperforms the binary bot or human classifier BBC. In the next figure, see Figure 7, we present the distribution of maximum prediction probabilities for the whole dataset. Apparently, apart from a few cases where the maximum probability lies below the 0.5 threshold, most instances have been correctly classified with a high probability.

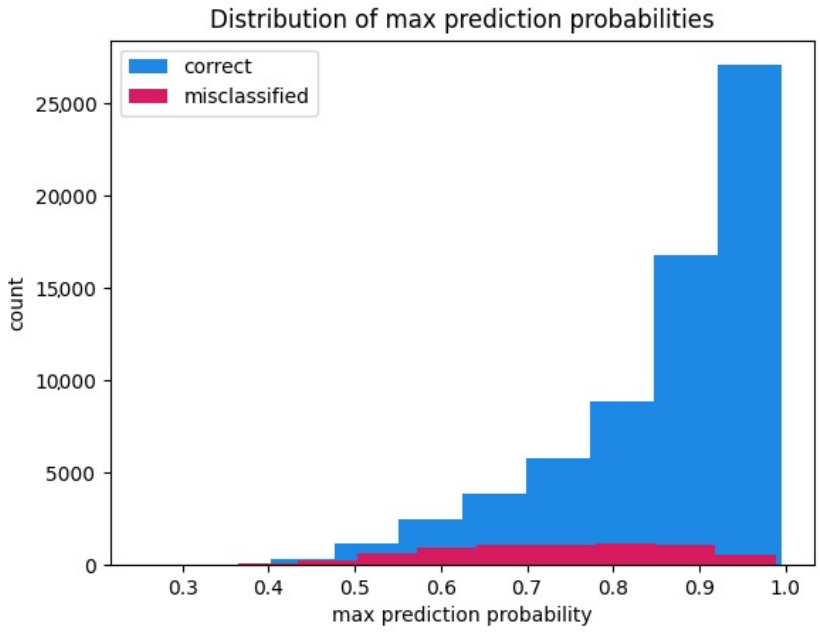

Figure 7. Distribution of maximum prediction probabilities for both correct and misclassified instances. Higher values show that our model predicts the instance class with higher confidence. We observe that the largest part of the correctly classified instances, were predicted with high confidence.

In this case, defining the most important features is not trivial, since we refer to a multi-class classification problem. For this reason, we examined the most important features per bot type, using the binary bot type classifiers. The results once more validated our hypothesis and manual assignment of labels to each bot type. More specifically, our experiments showed that the top 25 features are completely different for each bot type. This means, that there is not even one single feature among the twenty five top-ranking ones that is present in every class. In Figure 8, we observe that, surprisingly, the important features tend to become common for all bot classes after we reach the top 250, while the first common one makes its appearance after the first 25 ranked as most important.

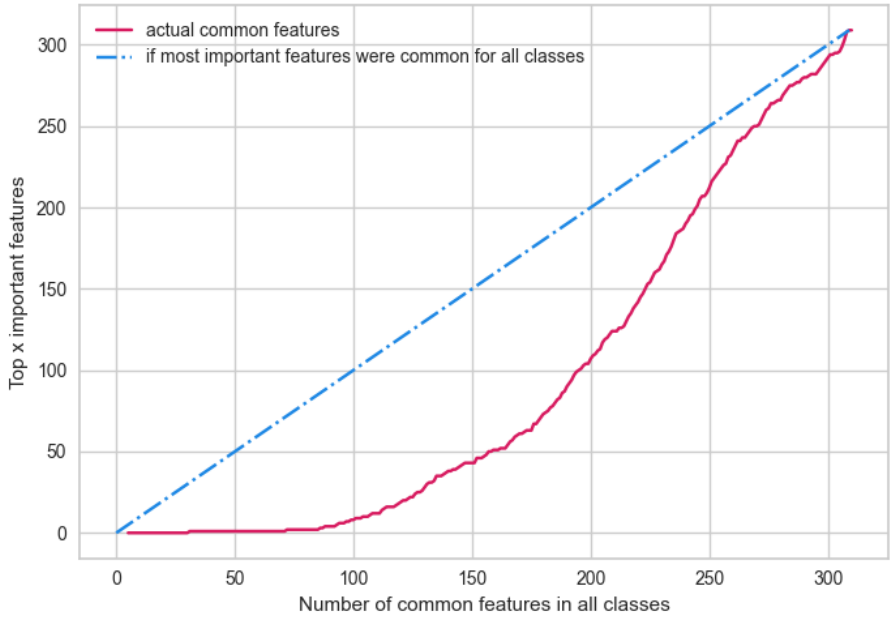

Figure 8. The union of the 25 most important features per bot type is equal to zero. Common, for all classes, important features make their appearance after the first 25 most important ones.

Regarding the type of features that have been found to be most important for each class, Table 8 summarizes our findings.

Table 8. Distribution of the 20 most important feature types for each bot type. Content features play an important role in almost every bot type.

Bot Type	User	Content	Temporal	Sentiment	Hashtag Corr
Spambot	4	12	-	4	-
Socialbot	1	18	1	-	-
Cyborg	4	11	5	-	-
Self-declared	3	12	-	-	5
Political	5	5	10	-	-

More specifically, we analyzed the twenty most important features per bot type and we discovered that, unlike in the case of the binary bot classifier, content features play the most important role in distinguishing humans from specific types of bots. It is also interesting that besides the fact that content features are prevalent for all bot types, none of those ranked as more important are common for all. Taking into consideration the wide range of improvements of this new model, with respect to the one already available in our bot-detection service, namely Bot-Detective, we have decided to use it in the updated beta version of our publicly available web service.

6. Explainability and Statistical Analysis

As highlighted in our previous work [58], targeting bot detection without explainable functionality is prohibiting in decision rationalization and trustfulness. In this work, the predictions of our models are augmented with highlighting the features that define whether an account is a human or a bot and provide the reasoning behind this prediction. Most importantly, in the multi-class detection problem, this strategy showcases the differences in each bot class, to reflect the behavior of accounts belonging in each class.

This work proposes a simple and flexible explainability pipeline (outlined in Figure 9). Although a Random Forest classifier provides interpretable results by design, its explain-

ability can not be utilized in our case due to the large number of features and number of estimators used. In our explainability pipeline we use the widely accepted LIME [66] as our explainer, to offer interpretable predictions of the binary classifier or/and the multi-class classifier.

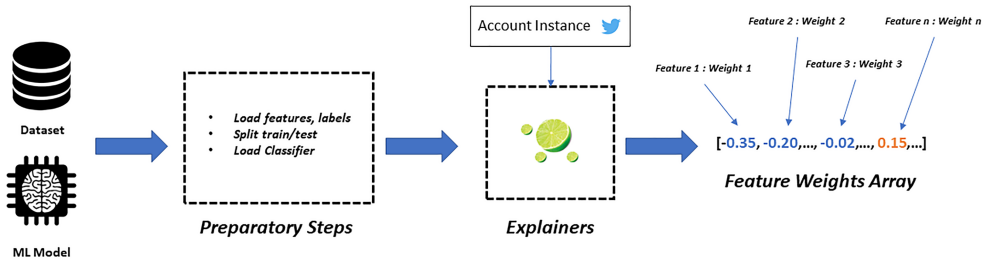

Figure 9. The explainability workflow.

The explainability phase, as depicted in Figure 1, resulted in explainer predictions for each instance, accompanied with the features that affected this prediction, all adjusted in the framework of the model that we were examining per explainer. For example, the explanations of the baseline binary classifier would have a "bot" or "human" label predicted by the classifier, and interpreted by the explainer, a true label of the same value which can obviously be close to the found label and the top ten features with their corresponding weights. Even though the results of the explainers are quite useful in not only validating the feature engineering process, but also shedding a clear light on what differentiates bot classes, they still remain static representations and lack a graphical visualization. To that end, once we completed the explaining process by conducting a statistical analysis and measurement in order to pinpoint the varying aspects of each class, distribution plots were utilized for the top features of each class belonging to the two classifiers that portrayed the distribution of values for every instance of the datasets. Moreover, we experimented with some key parameters of the explainability framework, in an effort to provide as realistic and accurate results, as possible. Our work towards this direction is further presented below.

Refining Explainable Results

Previous work on explaining the results of human–bot classifiers has been strictly based on providing some information on the features that define each prediction. Moreover, this analysis has only been applied to binary classifiers. In this section, we present our attempt to provide improved results, not only by experimenting with different parameters, but by extending the application to all bot types that are referred to in this paper.

We will use a popular explainability framework called LIME. Although LIME is considered to be a state-of-the-art tool for ML interpretability, recent research has shown that in some cases it faces some issues of instability [67,68], mainly due to the sampling step when new instances are randomly chosen. Improving the model is beyond the scope of this paper, however we experiment with different kernel widths, which seem to greatly affect the predictions made by LIME in comparison to those produced by our ML models, see Figure 10. Kernel width actually defines the region around the reference point from which new points are generated.

In order to provide more realistic explanations, we calculated the mean squared error (MSE) of our ML model predictions to those of LIME for different kernel widths. We can clearly observe that as the kernel width parameter increases, so does the mean square error. However, having a really low MSE is not always what should be preferred, since setting the kernel width to a very low value would mean that our main goal would be to predict this exact point correctly, which is not the case. Thereby, in our experiments we set the kernel width value equal to five, which seems to complement the MSE with the generalization of our model.

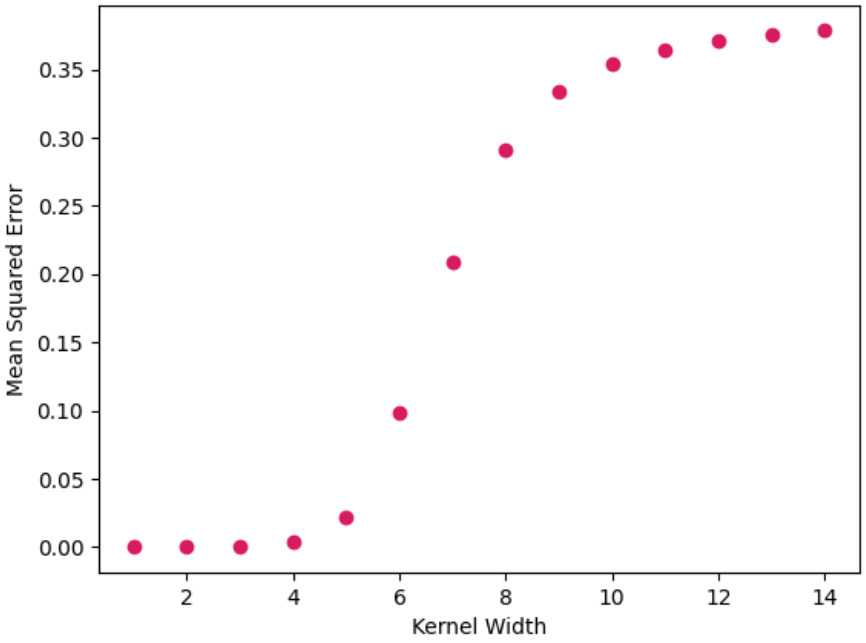

Figure 10. As the kernel width parameter increases, so does the mean square error.

Initially we applied our explainability model to the BBC classifier, in order to highlight the main features that affect our model's predictions. We created our explainer using the data that our model had been trained on and tested it on a sample of the spare test data. The next step involved applying a similar methodology for all the binary bot type classifiers, though limiting our dataset only to humans and certain bot types, depending on the classifier we were trying to interpret. The key findings are presented in Table 9, along with the most important features per bot class.

Table 9. Five most important features that defined the prediction of each bot type, according to our explainer model. The ratio refers to the number of times each feature was present in the five most important features for 200 instances.

All Bots	**Spambots**	**Social Bots**
Tweet retweet Ratio—49%	Tweet retweet Ratio—17%	Entropy avg time between tweets—16%
Followees Count—45%	min Text entropy—17%	Name screen similarity—14%
Median Text entropy—36%	Followees count—16%	triangles—13%
Mean Text entropy—35%	Followers count—15%	Min similarity—12%
Tweets per hour entropy—35%	entropy of avg time between tweets—14%	skewness of symbols—12%
Cyborgs	**Self-Declared**	**Political**
Followers count—60%	Entropy of tweets per hour—25%	Followees count—45%
Followers to followees—31%	entropy of Tweets per day—24%	entropy of avg time between tweets—25%
List count—27%	min avg time between tweets—16%	min text entropy—18%
Entropy of tweets per day—14%	min text entropy—13%	default profile—15%
Verified—14%	description length—13%	median of punctuation marks—13%

Finally, since our intentions included presenting an improved version of Bot-Detective, we decided to also present a visual explanation that rationalizes the prediction of our models, hence improving the interpretability of our results. To that end, our updated web service will also include figures which highlight the difference between the explored instance and the data that our model has been trained on. Such an example is presented in Figure 11, where the red line depicts the explored instance, while the histogram refers to the bots and humans included in our dataset.

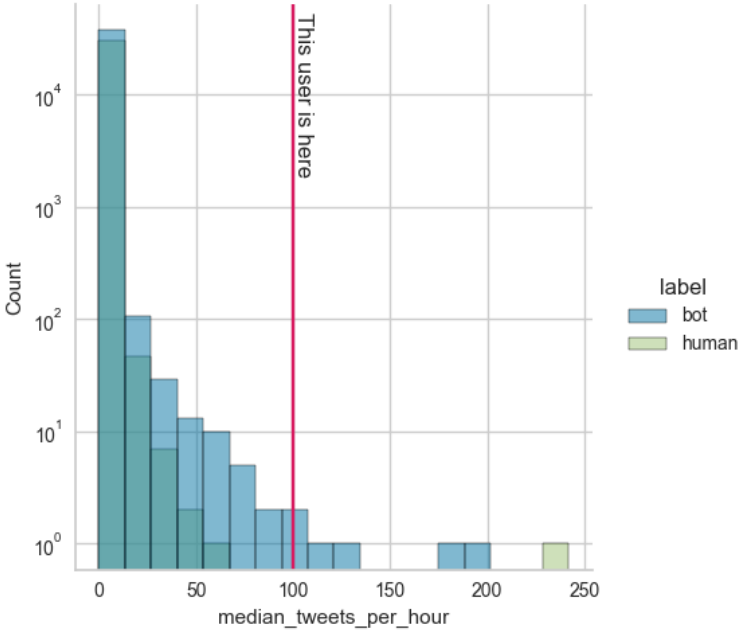

Figure 11. Visual explanation of where the explored user resides, with respect to the total training data. The red line points out where the user under investigation is.

7. Discussion

Our results and experimentation with supervised ML models and bot-related data towards effective bot detection are further discussed thoroughly in the next sections.

7.1. Data Exploration

The vast majority of published research on supervised ML for bot detection uses publicly available annotated data to train their models. They use specific datasets to highlight the efficiency of their approach. We have rehydrated all of the datasets and we discovered that a great part of them are outdated or include accounts that are no longer available in Twitter. More specifically, out of the 170 K accounts included, only 69 K are still available, out of which, the ones that are still active mainly belong to human users. Based on this fact, we can clearly come to the conclusion that the respective ML models are also outdated and do not follow a realistic approach. On the contrary our work is based on truly available data and takes a step further into distinguishing different bot classes among them. However, it is important to point out the never-ending need for new, up-to-date, credible annotated datasets and thus, due to the lack of such data, rationalize the scientific turn to unsupervised methods for bot detection.

7.2. Feature Selection

During the past years, the type and number of features that have been used for training has been constantly growing. As presented by other researchers, this number has exceeded 1200 single features. Taking into consideration that previously available information is no longer accessible (e.g., followers growth rate per tweet), highlights the need for constantly updated feature sets. Moreover, instead of persistently increasing the number of features, a wiser choice would be to assess their importance and try to achieve a balance between efficiency and accuracy. Towards that direction, our work has shown that a careful selection of features can still provide comparable results for binary bot or human detection. However, this is not the case in the multi-class bot classification problem, where the important features vary for each bot class.

7.3. Generalization and Accuracy

Previous research, our past work included, have reported really high accuracy in binary classification tasks, reaching a level of 98–99% in distinguishing bots from humans. However, most of this work has used specific datasets, which include all the labeled accounts, along with their respective tweets at the time of data collection. It is needless to say, that all these datasets have a high homogeneity score and such accuracy scores are not realistic, since, as also shown in [9], these models perform well only on the datasets that they have been trained on. On the contrary, we train our binary model to the dataset that is produced by merging all available datasets, while removing the accounts that are no longer available in Twitter. Naturally, we do not expect to have a higher accuracy than other models that use more "biased" or richer data than ours. Nonetheless, our model performs well, achieving relatively high accuracy, precision and recall scores for previously unseen data, as presented in Section 4.2.

With respect to the multi-class bot classification task, we propose a set of multi-class classifiers trained on all the available datasets and with different labels than those proposed by a few works by other researchers. In Figure 12, we show the t-SNE plot, where we visualize our high dimension dataset, limited for user features, in two dimension plots and show the clusters formed for social bots and cyborgs. We can clearly observe the distinction between these two classes.

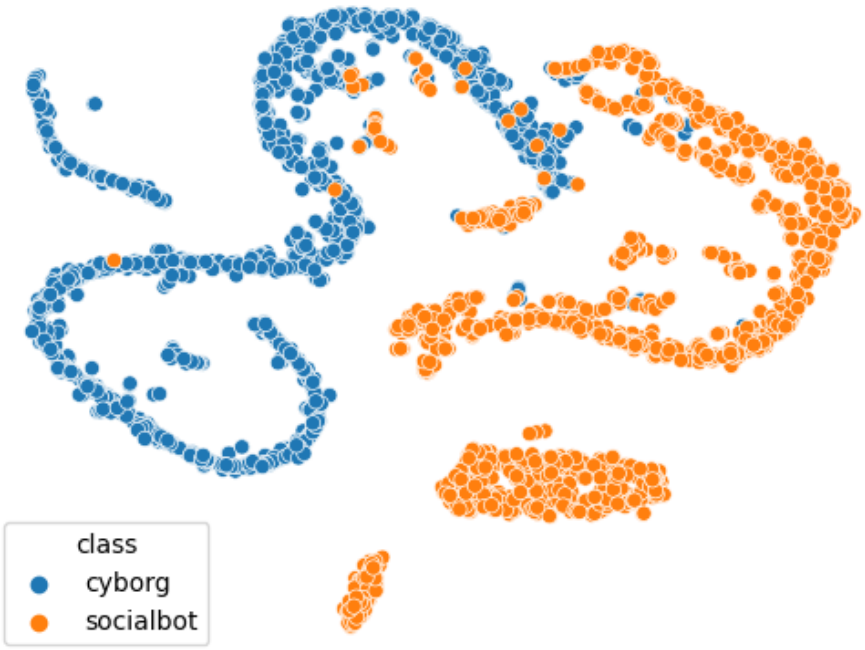

Figure 12. 2-D visualization of our high dimensional dataset, limited to user features for cyborgs and social bots. The distinction between these two classes is clear.

A direct comparison between ours and other approaches is not feasible, due to the different methodology followed during the training phase and the different training data. However, after taking into consideration the performance metrics that were reported in [9], where they achieve an average recall score of 0.84 and F1-score of 0.73, our method seems to provide improved results with an average recall score of 0.92 and F1-score of 0.9 for our best solution, namely the EBBC classifier. Finally, we attempt to improve the explainability of our predictions by providing more realistic and accurate results, as shown in Section 6.

Finally, we should highlight that our approach, with some small modifications mostly on the features used for the training of the models, could be used for the detection of bots in other OSNs (such as Facebook, Instagram, etc.). Nevertheless we focus on Twitter bots, since Twitter is the only OSN that offers an open API for data collection.

8. Conclusions and Future Work

In this paper we investigated the current state of supervised ML bot detection approaches on Twitter. We showed that the available data are mostly outdated and that previous research does not comply with the present data status, but rather focuses on past data. We also proposed a new bot type classification schema, based on the descriptions of the publicly available annotated datasets and newly introduced ones and proved its efficiency.

We perform a comprehensive feature analysis, enriched with explainability functionalities and demonstrate that different features account for different type of bots. Although we acknowledge the drawbacks of these data, we follow a different methodology to provide novel models for binary and multi-class bot detection. Our experiments show that our models perform really well on previously unseen data and that they generalize well, since they are tested on data coming from different datasets.

In future work, taking into consideration the lack of credible, up-to-date data, we intend to investigate adversarial methods to improve our models and make them adaptive to future, currently unobserved, new type of bots. Our main future goal would be to create novel generative models able to produce adversarial bot examples, leaving out the issue of data unavailability, and testing new discriminators against old and newly produced types of bots.

Author Contributions: Conceptualization, I.D. and A.V.; data curation, I.D. and K.G.; formal analysis, I.D.; funding acquisition, I.D.; investigation, I.D. and K.G.; methodology, I.D.; project administration, A.V.; resources, I.D. and A.V.; software, I.D.; supervision, A.V.; validation, I.D. and K.G.; vizualization, I.D.; writing—original draft, I.D.; writing—review and editing, I.D., K.G. and A.V. All authors have read and agreed to the published version of the manuscript.

Funding: This work was co-financed by the European Union and Greek national funds through the Operational Program for Competitiveness, Entrepreneurship and Innovation, under the call RESEARCH | CREATE | INNOVATE (Project Code: T2EDK-03898), as well as from the H2020 Research and Innovation Programme under Grant Agreement No.875329.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: We publicly share the IDs of Twitter accounts labeled as News Agencies and Organizations at https://github.com/idimitriadis/Botomics (accessed on 1 October 2021).

Conflicts of Interest: The authors declare no conflict of interest

Abbreviations

The following abbreviations are used in this manuscript:

OSN	Online Social Networks
ML	Machine Learning

References

1. Brossard, D.; Scheufele, D.A. Science, new media, and the public. *Science* **2013**, *339*, 40–41. [CrossRef]
2. Stieglitz, S.; Brachten, F.; Ross, B.; Jung, A.K. Do social bots dream of electric sheep? A categorisation of social media bot accounts. *arXiv* **2017**, arXiv:1710.04044.
3. Guglielmi, G. The next-generation bots interfering with the US election. *Nature* **2020**, *587*, 21–21. [CrossRef] [PubMed]
4. Lee, K.; Eoff, B.; Caverlee, J. Seven months with the devils: A long-term study of content polluters on twitter. In Proceedings of the International AAAI Conference on Web and Social Media, Barcelona, Spain, 17–21 July 2011; The AAAI Press: Menlo Park, CA, USA, 2011.
5. Yang, K.C.; Varol, O.; Davis, C.A.; Ferrara, E.; Flammini, A.; Menczer, F. Arming the public with artificial intelligence to counter social bots. *Hum. Behav. Emerg. Technol.* **2019**, *1*, 48–61. [CrossRef]
6. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. The paradigm-shift of social spambots: Evidence, theories, and tools for the arms race. In Proceedings of the WWW '17: 26th International World Wide Web Conference, Perth, Australia, 3–7 April 2017; pp. 963–972.

7. Bessi, A.; Coletto, M.; Davidescu, G.A.; Scala, A.; Caldarelli, G.; Quattrociocchi, W. Science vs conspiracy: Collective narratives in the age of misinformation. *PLoS ONE* **2015**, *10*, e0118093. [CrossRef] [PubMed]
8. Yang, K.C.; Varol, O.; Hui, P.M.; Menczer, F. Scalable and generalizable social bot detection through data selection. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 1096–1103.
9. Sayyadiharikandeh, M.; Varol, O.; Yang, K.C.; Flammini, A.; Menczer, F. Detection of novel social bots by ensembles of specialized classifiers. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Virtual Event, 19–23 October 2020; pp. 2725–2732.
10. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. Fame for sale: Efficient detection of fake Twitter followers. *Decis. Support Syst.* **2015**, *80*, 56–71. [CrossRef]
11. Ratkiewicz, J.; Conover, M.; Meiss, M.; Gonçalves, B.; Flammini, A.; Menczer, F. Detecting and tracking political abuse in social media. In Proceedings of the International AAAI Conference on Web and Social Media, Barcelona, Spain, 17–21 July 2011; Volume 5.
12. Bessi, A.; Ferrara, E. Social bots distort the 2016 US Presidential election online discussion. *First Monday* **2016**, *21*. [CrossRef]
13. Chu, Z.; Gianvecchio, S.; Wang, H.; Jajodia, S. Detecting automation of twitter accounts: Are you a human, bot, or cyborg? *IEEE Trans. Dependable Secur. Comput.* **2012**, *9*, 811–824. [CrossRef]
14. Broniatowski, D.A.; Jamison, A.M.; Qi, S.; AlKulaib, L.; Chen, T.; Benton, A.; Quinn, S.C.; Dredze, M. Weaponized health communication: Twitter bots and Russian trolls amplify the vaccine debate. *Am. J. Public Health* **2018**, *108*, 1378–1384. [CrossRef]
15. Chatzakou, D.; Kourtellis, N.; Blackburn, J.; De Cristofaro, E.; Stringhini, G.; Vakali, A. Mean birds: Detecting aggression and bullying on twitter. In Proceedings of the 2017 ACM on Web Science Conference, Troy, NY, USA, 25–28 June 2017; pp.13–22.
16. Marlow, T.; Miller, S.; Roberts, J.T. Twitter Discourses on Climate Change: Exploring Topics and the Presence of Bots. 2020. Available online: https://osf.io/preprints/socarxiv/h6ktm/ (accessed on 13 October 2021).
17. Young, V.A. Nearly Half of the Twitter Accounts Discussing 'Reopening America' May Be Bots. 2020. Available online: https://www.cmu.edu/news/stories/archives/2020/may/twitter-bot-campaign.html (accessed on 1 October 2021)
18. Cresci, S.; Lillo, F.; Regoli, D.; Tardelli, S.; Tesconi, M. $FAKE: Evidence of spam and bot activity in stock microblogs on Twitter. In Proceedings of the International AAAI Conference on Web and Social Media, Stanford, CA, USA, 25–28 June 2018; Volume 12.
19. Cresci, S.; Lillo, F.; Regoli, D.; Tardelli, S.; Tesconi, M. Cashtag piggybacking: Uncovering spam and bot activity in stock microblogs on Twitter. *ACM Trans. Web* **2019**, *13*, 1–27. [CrossRef]
20. Savage, S.; Monroy-Hernandez, A.; Höllerer, T. Botivist: Calling volunteers to action using online bots. In Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing, San Francisco, CA, USA, 26 February–2 March 2016; pp. 813–822.
21. Luckerson, V. Can Twitter Solve Its Big, Bad Bot Problem? 2018. Available online: https://www.theringer.com/tech/2018/3/8/17093982/twitter-bot-problem (accessed on 1 October 2021).
22. Cresci, S. A decade of social bot detection. *Commun. ACM* **2020**, *63*, 72–83. [CrossRef]
23. Twitter Co-Founder Jack Dorsey Answers Twitter Questions from Twitter | Tech Support | WIRED. 2020. Available online: https://youtu.be/de8wRd2TQQU?t=99 (accessed on 1 October 2021).
24. Conger, K. Twitter, in Widening Crackdown, Removes over 70,000 QAnon Accounts. 2021. Available online https://www.nytimes.com/2021/01/11/technology/twitter-removes-70000-qanon-accounts.html (accessed on 1 October 2021).
25. Craig Timberg, E.D. Twitter Is Sweeping Out Fake Accounts Like Never before, Putting User Growth at Risk. 2019. Available online: https://www.washingtonpost.com/technology/2018/07/06/twitter-is-sweeping-out-fake-accounts-like-never-before-putting-user-growth-risk/ (accessed on 1 October 2021).
26. Hutchinson, A. Twitter Says That Its Getting Better at Detecting and Removing Bots, Outlines Common Misinterpretations. 2020. Available online: https://www.socialmediatoday.com/news/twitter-says-that-its-getting-better-at-detecting-and-removing-bots-outlin/578272/ (accessed on 1 October 2021).
27. Tsvetkova, M.; García-Gavilanes, R.; Floridi, L.; Yasseri, T. Even good bots fight: The case of Wikipedia. *PLoS ONE* **2017**, *12*, e0171774. [CrossRef]
28. Varol, O.; Ferrara, E.; Davis, C.; Menczer, F.; Flammini, A. Online human-bot interactions: Detection, estimation, and characterization. In Proceedings of the International AAAI Conference on Web and Social Media, Montreal, QC, Canada, 15–18 May 2017; Volume 11.
29. Stella, M.; Ferrara, E.; De Domenico, M. Bots increase exposure to negative and inflammatory content in online social systems. *Proc. Natl. Acad. Sci. USA* **2018**, *115*, 12435–12440. [CrossRef] [PubMed]
30. Grinberg, N.; Joseph, K.; Friedland, L.; Swire-Thompson, B.; Lazer, D. Fake news on Twitter during the 2016 US presidential election. *Science* **2019**, *363*, 374–378. [CrossRef] [PubMed]
31. Gallotti, R.; Valle, F.; Castaldo, N.; Sacco, P.; De Domenico, M. Assessing the risks of 'infodemics' in response to COVID-19 epidemics. *Nat. Hum. Behav.* **2020**, *4*, 1285–1293. [CrossRef] [PubMed]
32. Nizzoli, L.; Tardelli, S.; Avvenuti, M.; Cresci, S.; Tesconi, M.; Ferrara, E. Charting the landscape of online cryptocurrency manipulation. *IEEE Access* **2020**, *8*, 113230–113245. [CrossRef]
33. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The rise of social bots. *Commun. ACM* **2016**, *59*, 96–104. [CrossRef]
34. Yardi, S.; Romero, D.; Schoenebeck, G. Detecting spam in a twitter network. *First Monday* **2010**, 15. v15i1.2793. [CrossRef]

35. Lee, K.; Caverlee, J.; Webb, S. Uncovering social spammers: Social honeypots+ machine learning. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 19–23 July 2010; pp. 435–442.
36. Yang, C.; Harkreader, R.C.; Gu, G. Die free or live hard? Empirical evaluation and new design for fighting evolving twitter spammers. In *International Workshop on Recent Advances in Intrusion Detection*; Springer: Berlin, Germany, 2011; pp. 318–337.
37. Lin, P.C.; Huang, P.M. A study of effective features for detecting long-surviving Twitter spam accounts. In Proceedings of the 2013 15th International Conference on Advanced Communications Technology (ICACT), PyeongChang, Korea, 27–30 January 2013; pp. 841–846.
38. Mccord, M.; Chuah, M. Spam detection on twitter using traditional classifiers. In *International Conference on Autonomic and Trusted Computing*; Springer: Berlin, Germany, 2011; pp. 175–186.
39. Yang, C.; Harkreader, R.; Gu, G. Empirical evaluation and new design for fighting evolving twitter spammers. *IEEE Trans. Inf. Forensics Secur.* **2013**, *8*, 1280–1293. [CrossRef]
40. Cai, C.; Li, L.; Zengi, D. Behavior enhanced deep bot detection in social media. In Proceedings of the 2017 IEEE International Conference on Intelligence and Security Informatics (ISI), Beijing, China, 22–24 July 2017; pp. 128–130.
41. Fazil, M.; Abulaish, M. A hybrid approach for detecting automated spammers in twitter. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2707–2719. [CrossRef]
42. Davis, C.A.; Varol, O.; Ferrara, E.; Flammini, A.; Menczer, F. Botornot: A system to evaluate social bots. In Proceedings of the 25th International Conference Companion on World Wide Web, Montréal, QC, Canada, 11–15 April 2016; pp. 273–274.
43. Grimme, C.; Preuss, M.; Adam, L.; Trautmann, H. Social bots: Human-like by means of human control? *Big Data* **2017**, *5*, 279–293. [CrossRef]
44. Shao, C.; Ciampaglia, G.L.; Varol, O.; Yang, K.C.; Flammini, A.; Menczer, F. The spread of low-credibility content by social bots. *Nat. Commun.* **2018**, *9*, 1–9. [CrossRef]
45. Zago, M.; Nespoli, P.; Papamartzivanos, D.; Perez, M.G.; Marmol, F.G.; Kambourakis, G.; Perez, G.M. Screening out social bots interference: Are there any silver bullets? *IEEE Commun. Mag.* **2019**, *57*, 98–104. [CrossRef]
46. Cresci, S.; Petrocchi, M.; Spognardi, A.; Tognazzi, S. Better safe than sorry: An adversarial approach to improve social bot detection. In Proceedings of the 10th ACM Conference on Web Science, Boston, MA, USA, 30 June–3 July 2019; pp. 47–56.
47. Wu, B.; Liu, L.; Yang, Y.; Zheng, K.; Wang, X. Using improved conditional generative adversarial networks to detect social bots on Twitter. *IEEE Access* **2020**, *8*, 36664–36680. [CrossRef]
48. Ruan, X.; Wu, Z.; Wang, H.; Jajodia, S. Profiling online social behaviors for compromised account detection. *IEEE Trans. Inf. For. Secur.* **2015**, *11*, 176–187. [CrossRef]
49. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. Emergent properties, models, and laws of behavioral similarities within groups of Twitter users. *Comput. Commun.* **2020**, *150*, 47–61. [CrossRef]
50. Chavoshi, N.; Hamooni, H.; Mueen, A. Debot: Twitter Bot Detection via Warped Correlation. In Proceedings of the 2016 IEEE 16th International Conference on Data Mining (ICDM), Barcelona, Spain, 12–15 December 2016; pp. 817–822.
51. Mazza, M.; Cresci, S.; Avvenuti, M.; Quattrociocchi, W.; Tesconi, M. Rtbust: Exploiting temporal patterns for botnet detection on twitter. In Proceedings of the 10th ACM Conference on Web Science, Boston, MA, USA, 30 June–3 July 2019; pp. 183–192.
52. Jiang, M.; Cui, P.; Beutel, A.; Faloutsos, C.; Yang, S. Catching synchronized behaviors in large networks: A graph mining approach. *ACM Trans. Knowl. Discov. Data* **2016**, *10*, 1–27. [CrossRef]
53. Jiang, M.; Cui, P.; Beutel, A.; Faloutsos, C.; Yang, S. Inferring lockstep behavior from connectivity pattern in large graphs. *Knowl. Inf. Syst.* **2016**, *48*, 399–428. [CrossRef]
54. Hooi, B.; Song, H.A.; Beutel, A.; Shah, N.; Shin, K.; Faloutsos, C. Fraudar: Bounding graph fraud in the face of camouflage. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 895–904.
55. Grimme, C.; Assenmacher, D.; Adam, L. Changing perspectives: Is it sufficient to detect social bots? In *International Conference on Social Computing and Social Media*; Springer: Berlin, Germany, 2018; pp. 445–461.
56. Echeverrï£¡ a, J.; De Cristofaro, E.; Kourtellis, N.; Leontiadis, I.; Stringhini, G.; Zhou, S. LOBO: Evaluation of generalization deficiencies in Twitter bot classifiers. In Proceedings of the 34th Annual Computer Security Applications Conference, San Juan, PR, USA, 3–7 December 2018; pp. 137–146.
57. Samper-Escalante, L.D.; Loyola-González, O.; Monroy, R.; Medina-Pérez, M.A. Bot Datasets on Twitter: Analysis and Challenges. *Appl. Sci.* **2021**, *11*, 4105. [CrossRef]
58. Kouvela, M.; Dimitriadis, I.; Vakali, A. Bot-Detective: An explainable Twitter bot detection service with crowdsourcing functionalities. In Proceedings of the 12th International Conference on Management of Digital EcoSystems, Abu Dhabi, United Arab Emirates, 2–4 November 2020; pp. 55–63.
59. Loyola-González, O.; Monroy, R.; Rodríguez, J.; López-Cuevas, A.; Mata-Sánchez, J.I. Contrast pattern-based classification for bot detection on twitter. *IEEE Access* **2019**, *7*, 45800–45817. [CrossRef]
60. Rauchfleisch, A.; Kaiser, J. The False positive problem of automatic bot detection in social science research. *PLoS ONE* **2020**, *15*, e0241045. [CrossRef]
61. Cresci, S.; Di Pietro, R.; Petrocchi, M.; Spognardi, A.; Tesconi, M. Social fingerprinting: Detection of spambot groups through DNA-inspired behavioral modeling. *IEEE Trans. Dependable Secur. Comput.* **2017**, *15*, 561–576. [CrossRef]

62. Gilani, Z.; Farahbakhsh, R.; Tyson, G.; Wang, L.; Crowcroft, J. Of bots and humans (on twitter). In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2017, Sydney, Australia, 31 July–3 August 2017; pp. 349–354.
63. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–8 June 2008; pp. 1322–1328.
64. Batista, G.E.; Bazzan, A.L.; Monard, M.C. Balancing Training Data for Automated Annotation of Keywords: A Case Study. In Proceedings of the II Brazilian Workshop on Bioinformatics, Macaé, Brazil, 3–5 December 2003; pp. 10–18.
65. Batista, G.E.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD Explor. Newsl.* **2004**, *6*, 20–29. [CrossRef]
66. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
67. Visani, G.; Bagli, E.; Chesani, F. OptiLIME: Optimized LIME Explanations for Diagnostic Computer Algorithms. *arXiv* **2020**, arXiv:2006.05714.
68. Alvarez-Melis, D.; Jaakkola, T.S. On the robustness of interpretability methods. *arXiv* **2018**, arXiv:1806.08049.