applied system innovation

MDPI

Article

An Explainable Feature-Based Approach for Understanding Social Bots Behaviour

Salvador Lopez-Joya [1,2,*], Jose A. Diaz-Garcia [1,3,*], M. Dolores Ruiz [1] and Maria J. Martin-Bautista [1]

[1] Department of Computer Science and AI, University of Granada, C. Periodista Daniel Saucedo Aranda, 18014 Granada, Spain; mdruiz@decsai.ugr.es (M.D.R.); mbautis@decsai.ugr.es (M.J.M.-B.)
[2] Research Centre for Information and Communications Technologies, C. Periodista Rafael Gómez Montero, 18014 Granada, Spain
[3] Department of Computer Languages and Systems, University of Granada, C. Cortadura del Valle s/n Ceuta, 51001 Ceuta, Spain
* Correspondence: slopezjoya@ugr.es (S.L.-J.); joseangeldiazg@ugr.es (J.A.D.-G.)

Abstract

The increasing influence of social media has amplified the risks associated with automated accounts that spread misinformation, manipulate public opinion and carry out malicious activities. To address this challenge, this study presents an explainable, feature-based approach for detecting social bots on 𝕏 (formerly Twitter) using user-profile information derived from account metadata and content characteristics. We consolidate and extend existing research by bringing together one of the most comprehensive feature sets explored to date, combining raw attributes, features proposed in the literature, and newly introduced credibility and engagement indicators, together with a previously unexploited profile-personalisation signal. Through a feature engineering and selection process that integrates Mutual Information, Random Forest Importance, and SHAP values, we evaluate the contribution of each feature category and assess its generalisation capacity across three benchmark datasets. Our experiments demonstrate that classical machine learning models enriched with the selected features can match or surpass several state-of-the-art approaches while preserving interpretability. Furthermore, we propose and validate, on the more recent and challenging TwiBot-22 dataset, three categories of features (universal, common, and dataset-specific) that provide a transparent and adaptable basis for generalisable bot detection.

Keywords: bot detection; feature engineering; machine learning; misinformation; social media analysis; explainable AI

Academic Editor: Wu Deng

Received: 20 May 2026
Revised: 25 June 2026
Accepted: 8 July 2026
Published: 10 July 2026

1. Introduction

The proliferation of social networks has undeniably transformed our daily lives, becoming an integral part of how we communicate with family and friends, a source of information on a wide range of topics [1], a platform for work, and a means of entertainment. This remarkable success, however, has also opened the door to harmful practices, such as the deliberate dissemination of misinformation. Many countries have voiced concerns about foreign interference in their electoral processes and social movements, frequently orchestrated by other states or organisations [2–5]. A significant share of this disinformation is spread by social bots, automated accounts that mimic human behaviour on social networks, generating and sharing content while interacting with unsuspecting users who are typically unaware that they are engaging with artificial entities. Detecting

the activity of these bots is critical to safeguarding the integrity of online information and preserving the authenticity of public discourse [6]. Their presence can also damage online ecosystems through malicious activities such as spamming, phishing, and cyber attacks [7,8].

Effective bot detection plays a crucial role in safeguarding online platforms, creating a secure and reliable environment for users. However, constant news reports (https://www.theguardian.com/us-news/2024/feb/26/ai-deepfakes-disinformation-election, https://abcnews.go.com/Politics/pro-trump-bots-sowing-division-republican-party-report/story?id=97997613, https://www.bloomberg.com/opinion/articles/2023-11-03/2024-campaign-don-t-let-chinese-bots-influence-the-next-us-election?embedded-checkout=true (accessed on 17 January 2026)) about the presence of bots in various aspects of people's lives suggest that there is still work to be done in the area of bot detection. Artificial Intelligence has emerged as one of the most promising avenues to address this challenge [9–11].

It is worth noting that, although the platform formerly known as Twitter has been rebranded to 𝕏, its core user behaviours (followers, posts, retweets, profile metadata) remain largely unchanged, so most of the features developed for bot detection on Twitter continue to be directly applicable on 𝕏. At the same time, 𝕏 has introduced new dynamics (monetisation, LLM-powered bots, API restrictions) that are not fully reflected in older benchmark datasets. For this reason, in this work we evaluate our approach not only on the historical benchmarks Cresci-15, Cresci-17 and TwiBot-20, but also on the larger and more recent TwiBot-22 dataset, which captures more contemporary bot dynamics and is, to date, the closest available proxy to the current 𝕏 ecosystem.

Exploring the literature, we can see two primary avenues of research in the realm of bot detection based on AI systems: one rooted in graph theory and network metrics, and the other centred on account-based and content-based metrics. Numerous authors have addressed the issue of bots in social networks in diverse domains such as public health, politics, and stock markets [12–14]. These authors propose novel approaches, predominantly defined or guided by characteristics related to account behaviour or content. Motivated by the premise that bots can be defined by their characteristics, this paper focuses on leveraging both account- and content-based features. We understand that the combination of these two types of features encompasses the term *user-profile* features. This study introduces a thorough feature engineering process to combat bots by leveraging user-profile measures. Our research aims to answer the following questions:

RQ. 1: What features define a social bot?
RQ. 2: Which source of features holds greater importance in social bot detection, account-based or content-based features?
RQ. 3: Can a social bot be identified based on user-profile features? Are they enough?

To answer the research questions, we have designed an experimental framework over four different datasets. While individual techniques such as Mutual Information, Random Forest Importance and SHAP are well established, our contribution lies in their systematic integration over an unusually broad feature space and across multiple datasets, together with the introduction of two novel groups of features and a generalisation-oriented categorisation of feature sets. In summary, our paper contributes significantly to the current state-of-the-art in several ways:

- We conduct a comprehensive review and consolidation of features proposed in the literature for social bot detection. To the best of our knowledge, this paper provides the most extensive analysis to date, combining a wider range of literature-based features and testing them on a wider range of benchmark datasets.

- We introduce two new families of user-profile features: a profile-personalisation set based on the categorisation of hexadecimal profile colours (colour binning), and two general-purpose social-interaction measures (credibility and engagement) adapted from [15].
- We provide a multi-perspective explainable analysis using three complementary techniques (Mutual Information, Random Forest Importance and SHAP values). Each technique captures a different aspect of feature relevance: non-linear dependence, ensemble-based interaction importance, and game-theoretic attribution. To the best of our knowledge, this is the most thorough and complete analysis outlining the features that affect the categorisation and classification of social bots.
- We propose a generalisation-oriented categorisation of features into *universal*, *common*, and *dataset-specific* sets, and validate this categorisation on TwiBot-22 [16], a more recent and considerably larger dataset. This provides an adaptable framework for future research and practical deployments, reducing reliance on platform-specific or black-box methods.
- We surpass several state-of-the-art non-graph-based methods on three benchmark datasets using classical machine learning algorithms, preserving interpretability and keeping computational cost moderate.

The structure of the paper is organised as follows: Section 2 provides an in-depth exploration of related works in the field. Our proposed framework for enhancing bot detection through feature engineering is described in Section 3. The experimental process is comprehensively detailed in Section 4. The subsequent section, Section 5, evaluates and interprets the results. Final remarks and potential extensions are considered in Section 6.

2. Related Works

This section aims to provide context and some of the related work in the literature. It starts with an introduction to the concept of a bot in social media, followed by a categorisation of feature-based bot detection methods, and finally an in-depth look at feature engineering and selection for bot detection.

2.1. Bots in Social Media

Although the authors generally agree that a bot in social networks is an account with a certain degree of automation, there is no extended definition that covers all the details related to these accounts. This is due to the speed at which technology advances and the doubts that exist when attributing certain characteristics to a bot. One of these characteristics is the level of automation required for an account to transition from a human-managed account to a bot; there are accounts that are partially automated, and establishing a threshold to differentiate between accounts that are not bots and accounts that are bots is complex. Examples of this can be seen in the work conducted by Pastor et al. [17], where they carried out a thorough experimentation focused on profiling bots according to their level of automation and behaviour across social media platforms, utilising social network analysis and graph theory.

Another of these features is the similarity to human behaviour. Some authors pay particular attention to this aspect by defining a bot in a social network context as an account that attempts to mimic human behaviour to a greater or lesser extent [18]. The last point to consider is the different fields of study from which these bots are studied; computer scientists tend to give more importance to more technical characteristics, while social scientists focus more on the social implications [19].

Depending on how we value these characteristics, we can give a more lax or restrictive definition. For example, as mentioned above, ref. [18] considers a social media bot is any

program that acts in the same way as a person in a social space, ref. [20] considers a bot as any account controlled by software within the social media, others like [21,22] emphasize that this account can be only partially automated. Among the most restrictive definitions, we can find the one given by [23] that considers a social media bot as a program that interacts with humans in a social environment and produces content automatically, adding that its intention is to mimic and perhaps alter human behaviour.

The definition followed in this study, as stated in [24], is as follows: *"a Social Media Bot is an account that is automated enough to produce content and/or interact with other accounts within a social media context."*

2.2. Feature-Based Bot Detection

There are three types of approaches for bot detection in the literature: feature-based, graph-based and crowdsourcing techniques. The most popular bot detection approaches are feature-based methods. Feature-based methods attempt to leverage the data contained in both the account metadata and the user-written text itself. Most methods based on deep learning and machine learning techniques fall into this category. Notably, this paradigm extends beyond social networks into the broader framework of cybersecurity and network traffic, where advanced AI models are utilized to differentiate legitimate user traffic from automated bot actions and malicious botnet activities [25]. Within the scope of social media platform analysis, these feature-based methods are typically divided into three categories: account-based, content-based and hybrid [24].

- **Account-based.** Account-based bot detection techniques use the information from the user's account as features or to infer new ones, e.g., account age, username length, number of retweets, number of followers, or follower growth rate. An example of such a method is given in [26]. In this study, the authors use only account-based features, making use of feature engineering and feature selection techniques. They provide a hybrid deep learning architecture divided into two layers, one for the most relevant numerical variables and another for the description of the profile using embeddings. This study achieves good generalisation and competitive results compared to other baselines.
- **Content-based.** Content-based bot detection techniques use information from the content of tweets as features, e.g., the number of URLs, the number of hashtags, the sentiment or the length of the tweet. An interesting example that falls into this category is presented in [27]. Their approach relies on analysing the temporal retweet activity among 𝕏 accounts. They employ an LSTM (A Long Short-Term Memory (LSTM) is a neural network belonging to the family of recurrent neural networks (RNNs). Unlike conventional RNNs, LSTM networks can learn short-term dependencies in sequential data while also possessing a long-term memory useful for learning broader dependencies.) variational autoencoder, a specialised neural network combining LSTMs' sequential data modelling with variational autoencoders' probabilistic distributions. This fusion allows the extraction of latent features from the retweet time series of individual accounts. Finally, they use a clustering algorithm for bot detection. Notably, this study stands out not only for its competitive results but also for its use of graphical representations. These graphs facilitate the visual exploration of the temporal patterns within each account, enhancing the interpretability of the results compared to other studies.
- **Hybrid.** Hybrid bot detection techniques use a combination of features from the user's account and its content. An example of a hybrid approach is highlighted in [28]. They used word embeddings from tweets, employing GloVe (Global Vectors) [29] and ELMo (Embeddings from Language Models) [30] for a contextualised semantic

representation of the text. Following this, they trained eight neural networks based on user profiling techniques, using characteristics such as gender, age, personality and education. By segmenting the dataset based on similar profiles, they enhanced classification accuracy. In the last step, the authors implemented a final model that has as input the values resulting from all previous models. Finally, to optimise the results, they explored different architectures for this final model, ultimately identifying the Feedforward Neural Network (FNN) as the most effective.

2.3. Feature Engineering and Feature Selection for Bot Detection

The process of creating, selecting, or transforming attributes or features that machine learning models use for making predictions is known as **feature engineering**. It involves extracting relevant information from the raw data and creating informative features that can improve the performance of the model [31,32].

In the context of social media bot detection, feature engineering involves designing features that effectively represent the unique behaviours and attributes of automated accounts. These may include:

- **Temporal patterns:** Metrics related to the frequency and timing of the user activity.
- **Social interactions:** Measures of the user's interactions, such as the number of followers, friends, and mentions.
- **Platform attributes:** Platform-dependent features, such as the presence of a profile picture, profile background colour or source of the user activity.
- **Language and content:** Features that describe the content and language used in tweets or posts, including sentiment analysis, stylometry, and linguistic complexity.
- **Network features:** Attributes related to the user's connections and network structure, such as centrality measures.

Effective feature engineering aids in identifying the underlying patterns that separate real users from automated programs. These features serve as a critical input to machine learning models and contribute to the ability of the model to accurately classify and detect bots.

In the same way, the process of choosing a subset of the most relevant features from the available feature set is known as **feature selection**. It is aimed at reducing dimensionality, improving model interpretability, and potentially enhancing model performance. Feature selection techniques are particularly valuable when working with high-dimensional datasets or when dealing with noisy and irrelevant features. Types of feature selection methods include:

- **Filter methods:** These methods evaluate the relevance of features independently of the machine learning model. Filter methods are efficient, but do not consider the relationship between features. Examples include Chi-Square [33], Mutual Information [34], and Fisher's Score [35].
- **Wrapper methods:** These methods use a machine learning model to evaluate different subsets of features. They include techniques such as Recursive Feature Elimination [36] and Forward/Backward Selection [37].
- **Embedded methods:** Feature selection is integrated into the model training process. These methods are able to capture feature dependencies. L1 regularisation [38] in linear models and Random Forest importance [39] are examples of embedded methods.

In the realm of bot detection leveraging feature engineering and selection, we found recent studies that overlap with our research [40–43].

In [40], Mbona and Eloff proposed leveraging Benford's Law to identify the most accurate features for bot categorisation and the development of a bot detection system.

They laid the foundation for further exploration in feature selection methodologies for bot categorisation, emphasising that a well-conceived approach for selection can yield superior results. However, the paper is predominantly focused on feature selection and does not provide a comparative analysis of their results in terms of bot identification against the established baselines in the literature. Moreover, it lacks the utilisation of benchmark datasets commonly employed in bot categorisation research.

Cardaioli et al. introduced the utilisation of writing style crafted features to improve bot categorisation and detection in their work [42]. Although the proposed features are interesting, their contribution is somewhat limited in terms of exploring interaction results among features, and their study focused solely on the Cresci-17 dataset, leaving cross-dataset generalisation untested. This is a limitation of single-dataset evaluation, not of the features themselves: several features we adopt come from such studies, and part of our contribution is to test their generalisation across multiple benchmarks. In contrast, the work presented in [41] introduces a deep-learning system that is fed with a comprehensive set of 66 newly crafted features encompassing both account-based and content-based aspects. The authors further proposed a meticulous feature engineering process, conducting a comparative set of experiments across two benchmark datasets, Cresci-17 and the Social Honeypot Dataset [44], achieving state-of-the-art results in both datasets.

The closest methodological precedent for our use of multiple feature-importance perspectives is found outside the bot-detection literature. Kim et al. [45] combine Shapley values and Mutual Information inside a multi-agent reinforcement-learning selector for medical post-operative outcome prediction, demonstrating that MI and SHAP capture complementary aspects of feature relevance. Our work differs in both setting (large-scale social-bot benchmarks rather than a single clinical dataset) and in how the two measures are used: we employ MI and TreeSHAP, together with Random Forest importance, as three diagnostic criteria over the same feature space rather than as components of a learned reward function.

Within bot detection, two recent works also incorporate SHAP-based explainability. Gupta and Singh [46] propose a three-layer framework that combines Recursive Feature Elimination, LightGBM/Random Forest classifiers and SHAP interpretation on the NCC-2 and CTU-13 network-traffic datasets, and Hossain et al. [47] embed SHAP into a federated-learning pipeline for IoT botnet detection through knowledge distillation across heterogeneous clients. Both works confirm that SHAP-augmented tree ensembles are an established methodological pattern in cybersecurity, but they target network traffic and IoT botnets rather than social media accounts, and neither attempts to extract a dataset-independent core of features. More directly relevant to our problem, Alhumaid et al. [48] use Random Forest together with network-analysis features and an explicit credibility score for social-bot identification. Their use of credibility as a discriminative signal aligns directly with the credibility/engagement feature family we propose in Section 3, although their approach additionally relies on network-structure features that our setting deliberately avoids, and they evaluate on a single dataset rather than test generalisation across benchmarks.

Compared to the four works above, our setting differs in scale, scope and aim. We apply MI, Random Forest importance and TreeSHAP jointly over more than 130 user-profile attributes, which is roughly an order of magnitude beyond the feature spaces considered in those studies. We evaluate generalisation on four datasets, with TwiBot-22 used only as a held-out validation set. And we use the agreement between the three measures to derive an explicit categorisation of features into universal, common and dataset-specific, which is not attempted in any of those papers.

Our research builds on this feature engineering methodology by merging features from various papers, as well as introducing two new features based on social interaction

and a set of features based on profile colour personalisation. Compared to previous studies that typically focus on a single dataset and a single importance metric, we jointly analyse three importance perspectives, extend the evaluation to three established benchmarks plus a more recent dataset, and explicitly target feature generalisation by defining universal, common and dataset-specific sets. This allows us to identify the most relevant features for detecting bots on social networks and evaluate the competitiveness of classical techniques based on user-profile features.

3. Enhancing Feature Engineering and Feature Selection for Bot Detection

A feature engineering approach has been followed to identify the features that are useful for differentiating genuine accounts from automated accounts. Three of the most widely spread datasets on bot detection have been chosen: Cresci-15 [49], Cresci-17 [50] and TwiBot-20 [51]. These datasets have been analysed by selecting useful features from them to serve as a basis for this process. The datasets do not have exactly the same features, so not all inferred features can be obtained in all datasets. A table with all the raw and inferred features, in which datasets they can be calculated and whether they have appeared in the literature can be found in our public repository (https://github.com/slopezjoya/An-Explainable-Feature-Based-Approach-for-Understanding-Social-Bots-Behaviour (accessed on 17 January 2026)). Three types of features of different natures have been obtained: raw features, literature features and proposed new features. The hierarchical diagram of these features can be seen in Figure 1.

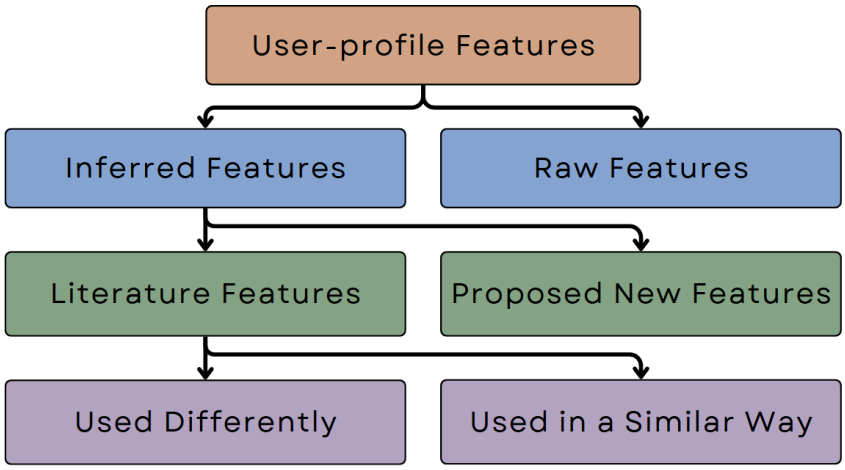


Figure 1. Feature engineering diagram.

Among the characteristics of the elements in each dataset, features that are inherently useful without requiring any inference process can be extracted. We have called these features *raw features* and, among them, we can find the number of followers, the number of favourites, the account verification or the number of lists in which the user appears. These characteristics will be the basis by which we can infer new ones.

From the *raw features*, new features have been inferred that could be of value for bot detection. These features are divided into features from the literature that have been explored by other authors and new features proposed in this study. We have calculated some of the classic meta-features of feature engineering, such as mean, ratios, minimum, maximum, among others, including them in the features coming from the literature if other authors have used them for the task of bot detection. An exhaustive study of the literature was carried out with the aim of finding calculable characteristics that other authors have proposed. Two different databases, Web of Science and Google Scholar, were used in this process. With 𝕏 as the focus, different terms and queries were used to perform the search,

including: *"bot detection in Twitter"*, *"bot detection feature engineering"*, *"identifying bots in Twitter"*, *"automated account detection in social media"*, *"bot characteristics"*, etc. The search results were ranked by relevance, and the top 20 from each query were selected when available. We reviewed each article and discarded those that did not introduce new features relevant to our problem. From the remaining papers, we chose those that proposed original and feasible features suitable for computation in our dataset. Semantic features were excluded to maintain interpretability, as they are often derived from deep learning models. Similarly, graph-based features were not considered due to the lack of network architecture data in many datasets and the high computational cost for large networks.

Tables 1 and 2 show the features derived from both the account and the content, grouped by type. In addition, they have been marked in the *Use* column with an *S* if they were used in the same fields as in the original study, or with a *D* if they were used in different fields.

Table 1. Inferred literature features from account. Italicised expressions in the Description column correspond to the formulas used to compute each feature.

Feature Name	Ref.	Description	Use	Type
Followers_growth_rate	[26]	$n\_followers/user\_age$	S	Social Based
Friends_growth_rate	[26]	$n\_friends/user\_age$	S	
Favourites_growth_rate	[26]	$n\_favourites/user\_age$	S	
Listed_growth_rate	[26]	$n\_listed/user\_age$	S	
Followers_friends_ratio	[26]	$n\_followers/n\_friends$	S	
Average_favorites	[52]	$n\_favorites/n\_followers$	S	
Average_retweets	[52]	$n\_retweets/n\_followers$	S	
Reputation	[41]	Reputation of the user	S	
User_age	[26]	The age of the account in days	S	Temporal
Tweet_freq	[26]	$n\_tweets/user\_age$	S	
Description_flesch_reading_ease	[42]	Flesch Reading Ease Score of description	D	Readability
Description_flesch_kincaid_grade	[42]	Flesch-Kincaid Grade of description	D	
Description_smog_index	[42]	SMOG index of description	D	
Description_coleman_liau_index	[42]	Coleman–Liau index of description	D	
Description_automated_readability_index	[42]	Automated Readability Index of description	D	
Description_dale_chall_readability_score	[42]	Grade level using the New Dale-Chall Formula in description	D	
Description_difficult_words	[42]	Number of difficult words in description	D	
Description_linsear_write_formula	[42]	Grade level using Linsear Write Formula of description	D	
Description_gunning_fog	[42]	Gunning fog index of description	D	
Screen_name_length	[26]	Length of screen name	S	Stylometry
Name_length	[26]	Length of name	S	
Description_length	[26]	Length of description	S	
Description_digits_count	[26]	Count of digits in description	D	
Description_mean_bigram_freq	[26]	Mean bigram freq. in description	D	
Screen_name_digits_count	[26]	Count of digits in screen name	S	
Name_digits_count	[26]	Count of digits in name	S	
Screen_name_mean_bigram_freq	[26]	Mean bigram freq. in screen name	S	
Screen_name_entropy	[26]	Entropy of screen name	S	
Name_mean_bigram_freq	[26]	Mean bigram freq. in name	D	
Name_entropy	[26]	Entropy of name	S	
Description_entropy	[26]	Entropy of description	S	
Name_sim	[26]	Name and screen name similarity	S	
Name_ratio	[26]	Name and screen name length ratio	S	
Name_contains_bot	[53]	If name contains "bot"	D	
Screen_name_contains_bot	[53]	If screen name contains "bot"	D	
Description_contains_bot	[53]	If description contains "bot"	D	
Description_hashtag_count	[41]	Hashtags count in description	S	

Table 1. *Cont.*

Feature Name	Ref.	Description	Use	Type
Description_url_count	[41]	URLs count in description	S	Stylometry
Description_unique_url_count	[41]	Unique URLs count in description	D	
Description_unique_mention_count	[41]	Unique mentions count in description	D	
Description_fraction_of_words_lowercase	[54]	Fraction of lowercase words in description	D	
Description_fraction_of_words_uppercase	[54]	Fraction of uppercase words in description	D	
Description_fraction_of_words_tilecase	[54]	Fraction of tilecase words in description	D	
Description_word_count	[54]	Number of words in description	D	
Description_sentence_count	[54]	Number of sentences in description	D	
Description_average_word_length	[54]	Average length of words in description	D	
Description_average_words_per_sentence	[54]	Description avg. words per sent.	D	

Table 2. Inferred literature features from content. Italicised expressions in the Description column correspond to the formulas used to compute each feature.

Feature Name	Ref.	Description	Use	Type
Ratio_retweet	[41]	*n_retweets*/*n_tweets*	S	Social Based
Average_time_between_tweets	[41]	Average time between tweets	S	Temporal
Idle_hours	[41]	Max time without activity	S	
Size_DNA_type	[55]	Size of DNA type before compression	S	
Compress_size_DNA_type	[55]	Size of DNA type after compression	S	
Compression_ratio_type	[55]	Ratio of DNA type sizes	S	
Size_DNA_content	[55]	Size of DNA content before compression	S	
Compress_size_DNA_content	[55]	Size of DNA content after compression	S	
Compression_ratio_content	[55]	Ratio of DNA content sizes	S	
Flesch_reading_ease	[42]	Average Flesch Reading Ease Score in tweets	S	Readability
Flesch_kincaid_grade	[42]	Average Flesch-Kincaid Grade in tweets	S	
Smog_index	[42]	Average SMOG index in tweets	S	
Coleman_liau_index	[42]	Average Coleman–Liau index in tweets	S	
Automated_readability_index	[42]	Average Automated Readability Index in tweets	S	
Dale_chall_readability_score	[42]	Average grade level using the New Dale-Chall Formula in tweets	S	
Difficult_words	[42]	Average number of difficult words in tweets	S	
Linsear_write_formula	[42]	Average grade level using Linsear Write Formula in tweets	S	
Gunning_fog	[42]	Average Gunning fog index in tweets	S	
Different_sources	[41]	*n_sources_used*/*n_total_sources*	S	Platform Based
Source_tweetadder_percentage	[56]	Percentage of tweets from tweetadder	S	
Source_iphone_percentage	[56]	Percentage of tweets from iphone	S	
Source_android_percentage	[56]	Percentage of tweets from android	S	
Source_twitter_percentage	[56]	Percentage of tweets from twitter	S	
Source_tweetdeck_percentage	[56]	Percentage of tweets from tweetdeck	S	
Source_ipad_percentage	[56]	Percentage of tweets from ipad	S	
Source_web_percentage	[56]	Percentage of tweets from web	S	
Source_facebook_percentage	[56]	Percentage of tweets from facebook	S	
Source_instagram_percentage	[56]	Percentage of tweets from instagram	S	
Source_api_percentage	[56]	Percentage of tweets from API	S	
Source_web_api_percentage	[56]	Percentage of tweets from web API	S	
Source_mobile_percentage	[56]	Percentage of tweets from mobile	S	
Source_other_percentage	[56]	Percentage of tweets from other	S	

Table 2. *Cont.*

Feature Name	Ref.	Description	Use	Type
Bot_reference_mean	[53]	References mean to "bot" in tweets	S	
Average_tweet_length	[41]	Average length of tweets	D	
Num_unique_urls_mean	[41]	Unique URLs count in tweets	D	
Num_unique_mentions_mean	[41]	Unique mentions count in tweets	D	
Max_urls_in_a_tweet	[41]	Max number of URLs in a tweet	S	
Max_hashtags_in_a_tweet	[41]	Max number of hashtags in a tweet	S	
Max_mentions_in_a_tweet	[41]	Max number of mentions in a tweet	S	
Average_tweets_only_url	[41]	Average tweets with one URL	S	
Average_elongated_words	[41]	Average elongated words in tweets	S	
Num_unique_langs	[41]	Number of unique langs in tweets	S	
Word_count_mean	[54]	Mean of word count in tweets	S	Stylometry
Sentence_count_mean	[54]	Mean of sentence count in tweets	S	
Average_word_length	[54]	Average length of words in tweets	S	
Average_words_lowercase	[54]	Average number of lowercase words in tweets	S	
Average_words_uppercase	[54]	Average number of uppercase words in tweets	S	
Average_words_titlecase	[54]	Average number of tilecase words in tweets	S	
Tweets_sim_length	[56]	Similarity of tweet lengths	S	
Tweets_sim_punctuation	[56]	Similarity of tweet punctuation	S	

3.1. Proposed New Features

Two new sets of features have been proposed. The first one is account-based and aims to take advantage of the information that can give us the level of personalisation of the user's account when detecting bots. In the second one, we recover the measures exposed in [15] that allow modelling the credibility and engagement of a user based on the metadata of the tweets. Tables 3 and 4 summarise the complete set of these features.

3.1.1. Colour Binning

Among the raw features that can be extracted from the $\mathbb{X}$ account are those that represent how the user has customised the profile using colours in hexadecimal base. To the best of our knowledge, these features have not been previously exploited in the literature. Our intuition is that a high degree of profile personalisation is more characteristic of human accounts, while bots (particularly those created in bulk) tend to either keep the platform defaults or reuse a small set of values. The features introduced in this section operationalise that intuition in a transparent and reproducible way.

Formally, let $\mathcal{C}_d$ denote the set of hexadecimal values observed for a given colour-related attribute d (e.g., *profile_sidebar_fill_color*) across all users of a dataset, and let $\mathrm{FREQ}(c)$ be the frequency with which value $c \in \mathcal{C}_d$ appears. Let c^d_{default} be the platform default value for d (as provided by $\mathbb{X}$ in the account metadata), and let $\mathcal{T}^d_8 \subseteq \mathcal{C}_d \setminus \{c^d_{\text{default}}\}$ be the set of the 8 most frequent non-default values. For each user u with observed value c^d_u, we define three binary indicator features:

$$\mathbb{1}^d_{\text{def}}(u) = \mathbb{1}\left[c^d_u = c^d_{\text{default}}\right], \tag{1}$$

$$\mathbb{1}^d_{\text{com}}(u) = \mathbb{1}\left[c^d_u \in \mathcal{T}^d_8\right], \tag{2}$$

$$\mathbb{1}^d_{\text{unc}}(u) = \mathbb{1}\left[c^d_u \notin \{c^d_{\text{default}}\} \cup \mathcal{T}^d_8\right]. \tag{3}$$

The three indicators are mutually exclusive by construction. The choice of $|\mathcal{T}^d_8| = 8$ is empirically motivated by the long-tail distribution of profile colours observed in the three datasets, illustrated in Figure 2: the top eight values consistently account for the bulk of non-default usage, beyond which usage frequency drops sharply. We performed a

sensitivity check varying this cut-off in the range $\{4, 8, 12, 16\}$ and found that the resulting feature importance and downstream accuracy were stable, with no statistically meaningful difference. We therefore retain 8 as a conservative round value. Figure 2 illustrates the procedure for the *profile_sidebar_fill_color* attribute.

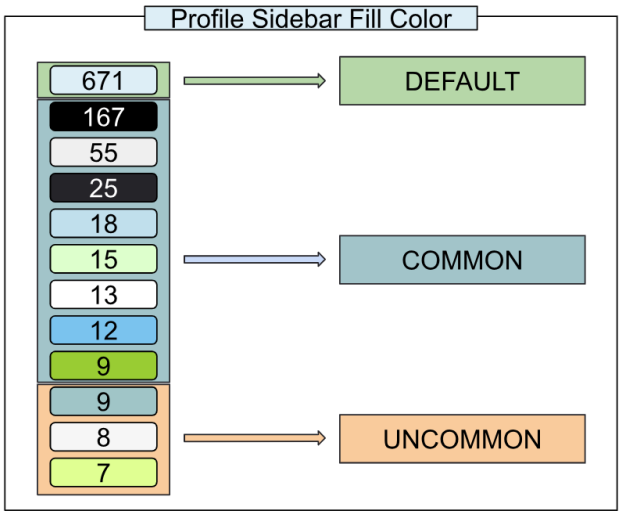


Figure 2. Profile sidebar fill colour ranking.

Table 3. Inferred new features based on user account.

Feature Name	Description	Type
Profile_background_color_is_default	If profile background colour is default	Platform Based
Profile_background_color_is_uncommon	If profile background colour is uncommon	
Profile_background_color_is_common	If profile background colour is common	
Profile_background_image_url_default_other_none	If profile background image exists, it is the default or other	
Has_profile_background_tile	If has profile background tile	
Profile_link_color_default	If profile link colour is default	
Profile_link_color_common	If profile link colour is common	
Profile_link_color_uncommon	If profile link colour is uncommon	
Profile_sidebar_border_color_default	If profile sidebar border colour is default	
Profile_sidebar_border_color_common	If profile sidebar border colour is common	
Profile_sidebar_border_color_uncommon	If profile sidebar border colour is uncommon	
Profile_sidebar_fill_color_default	If profile sidebar fill colour is default	
Profile_sidebar_fill_color_common	If profile sidebar fill colour is common	
Profile_sidebar_fill_color_uncommon	If profile sidebar fill colour is uncommon	
Profile_text_color_default	If profile text colour is default	
Profile_text_color_common	If profile text colour is common	
Profile_text_color_uncommon	If profile text colour is uncommon	

We introduce these indicators as a low-cost, previously unexploited personalisation signal whose value we assess empirically rather than assume a priori, as shown following sections, their contribution turns out to be modest and confined to datasets where the profile-colour metadata is populated.

3.1.2. Credibility and Engagement

In the study conducted by the authors in [15,57], a filter based on the assignment of credibility and knowledge of users on a specific topic is proposed. The main objective of this study is to reduce irrelevant content coming from social networks, thus allowing users to filter and highlight useful and credible content.

In this filter, engagement is mathematically modelled by relating the number of favourites and the number of retweets on a topic to the number of followers, to then establish a cut-off threshold. In the same way, they model credibility relating the number of followers, the number of listings, the number of retweets and the number of favourites to establish another cut-off threshold.

In this work, they do not use general engagement but engagement on a certain topic. For the authors, the fact that a person generates quality content on a specific topic does not mean that he/she has to do it for other topics. In the context of our study, we apply this formula in a general way, as our analysis does not delve into the problem at the topic level:

- Engagement:

$$\epsilon(u) = \frac{n\_Favorites + n\_Retweets}{2 \cdot n\_Followers} \tag{4}$$

  where $\epsilon(u)$ denotes the engagement of a user u, $n\_Favorites$ is the sum of favourites of the user's tweets, $n\_Retweets$ is the sum of retweets of the user's tweets, and $n\_Followers$ is the number of followers of the user. It is important to highlight that the division by the number of followers in Equation (4) acts as an intrinsic normalisation mechanism, equilibrating the metric across accounts with disparate follower scales. This approach ensures that the resulting engagement score reflects the relative intensity of user interaction rather than absolute volume. Consequently, a viral account with a massive following but low per-follower interaction will yield a lower engagement score than a smaller account with high community activity, which more accurately denotes content quality and user resonance.
- Credibility:

$$\zeta(u) = \frac{n\_Followers + n\_Lists + n\_Retweets + n\_Favorites}{4} \tag{5}$$

  where $\zeta(u)$ denotes the credibility of a user u, n_Lists is the number of public lists the user appears in, and $n\_Followers$, $n\_Retweets$ and $n\_Favorites$ are the same variables as in the engagement formula.

Regarding the presence of extreme outliers, although social media metrics typically exhibit a long-tail distribution, the primary objective of our formulas is to capture behavioural patterns through direct ratios. To ensure that these features do not disproportionately influence the classifier when encountering accounts with millions of followers, we implement a Min-Max normalization (as detailed in Section 4.2). This provides an additional robustness layer that maps all numerical inputs to a common range, effectively mitigating potential bias from high-reach accounts and ensuring the stability of the model's feature importance.

Table 4. Inferred new features based on user content.

Feature Name	Description	Type
Credibility	Credibility of the user	Social Based
Engagement	Engagement of the user	

4. Experiments

In this section, the datasets will be briefly described, as well as the methodology used. Following this, the results obtained will be presented, and it will conclude with an ablation study comparing the model using both sources of features: account and content.

4.1. Data

Cresci-17. This is a dataset of user accounts and tweets obtained from 𝕏 with the help of users of the CrowdFlower platform [50]. In Table 5, it is shown what they are and how the classes are distributed in the Cresci-17 dataset. Note that the classes *Traditional spambots #2*, *Traditional spambots #3* and *Traditional spambots #4* do not contain any data about tweets, as the authors considered them irrelevant for their work.

Table 5. Cresci-17 distribution [50].

Class Name	Accounts	Tweets
Genuine accounts	3474	8,377,522
Social spambots #1	991	1,610,176
Social spambots #2	3457	428,542
Social spambots #3	464	1,418,626
Traditional spambots #1	1000	145,094
Traditional spambots #2	100	74,957
Traditional spambots #3	433	5,794,931
Traditional spambots #4	1128	133,311
Fake followers	3351	196,027

Cresci-15. This dataset belongs to the same project as the previous one and focuses on detecting fake followers [49]. To collect data from legitimate accounts, the authors used 𝕏 and followed two approaches. The first involved creating an account named @TheFake-Project, with a bio stating: "Follow me only if you are NOT a fake." The second method relied on the hashtag #elezioni2013, gathering 𝕏 accounts that engaged in discussions about Italian politics under this tag. These accounts were then manually verified to confirm their legitimacy. Table 6 shows the dataset's class distribution.

Table 6. Cresci-15 distribution [49].

Class Name	Accounts	Tweets	Followers	Friends
TFP (human)	469	563,693	258,494	241,710
E13 (human)	1481	2,068,037	1,526,944	667,225
FSF (bot)	1169	22,910	11,893	253,026
INT (bot)	1337	58,925	23,173	517,485
TWT (bot)	845	114,192	28,588	729,839

TwiBot-20. This dataset was collected in 2020, and in [51] the authors explain in detail the user selection process. The authors want to maintain user diversity by using a strategy based on seed users. These seeds come from four different domains: politics, business, entertainment, and sports. To identify the bot accounts, they ran a crowdsourcing campaign, assigning five annotators to each account for more robust identification. In Table 7, we can see what the classes are and how the classes are distributed in this dataset.

Table 7. TwiBot-20 distribution.

Entity	N. Samples
Genuine Accounts	5237
Bot Accounts	6589
Tweets	3,348,819
Edges	3,371,617

This dataset is available in JSON format with a section for the account features, a list with the raw text of up to 200 tweets and to which domain the account belongs.

Cresci-15 and Cresci-17 datasets can be downloaded from the official Botometer repository (https://botometer.osome.iu.edu/bot-repository/), while the Twibot-20 dataset requires access to the authors of the corresponding paper.

4.2. Methodology of Experimentation

A rigorous methodology has been followed to provide valid and consistent results (see Figure 3). As explained above, a literature review has been performed in order to find as many relevant features as possible. For each dataset, we have implemented these features (see Tables 1 and 2) in addition to the new ones we have suggested (see Tables 3 and 4). Once the features related to hashtags, mentions, emojis and URLs were calculated, we proceeded to the elimination of these elements in the text. For language detection, we used the *FastText* library, specifically a language detection model trained on data from *Wikipedia*, *Tatoeba* and *SETimes* [58,59]. Prior to the experiments and considering the long-tail distribution characteristic of social media metrics—where accounts often exhibit extreme outliers ranging from millions of interactions to none at all—we implemented a robust preprocessing pipeline. To mitigate potential bias and ensure that viral accounts do not disproportionately influence the classifier, all numerical features were transformed using Min-Max normalization to the range. This scaling method was selected over standardization to preserve the natural lower bound at zero, which is essential for the interpretability of social media ratios and growth rates. The transformation was applied independently to each feature and computed strictly on the training partition of each cross-validation fold to prevent information leakage, thereby ensuring the stability and generalizability of the model's results. Binary and one-hot encoded features were left unchanged. We chose Min-Max scaling rather than standardisation because many of our features (counts, ratios, growth rates) are highly skewed and not approximately Gaussian, and because Min-Max preserves the natural lower bound at zero that is meaningful for these variables.

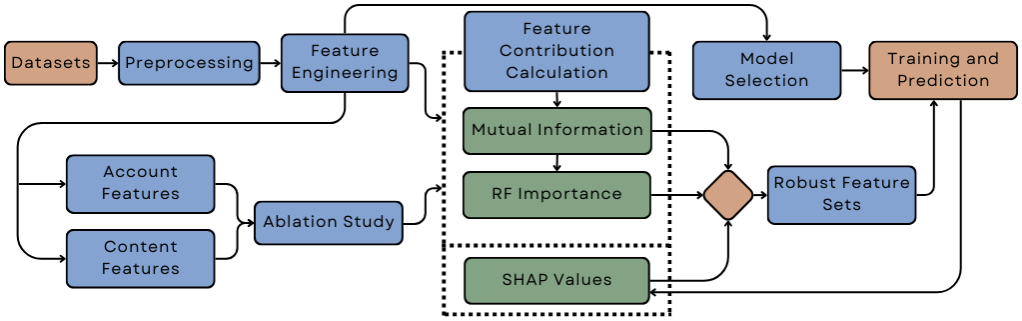


Figure 3. Experimentation flowchart.

Prior to model training, we addressed the characteristic long-tail distributions and extreme outliers inherent to social media metrics, where account activity can span several orders of magnitude (ranging from millions of interactions to none). While logarithmic smoothing is frequently employed to compress such power-law distributions, we intentionally avoided it to preserve the raw semantic interpretability and direct proportionality of our engagement and credibility metrics. Instead, to mitigate potential bias and ensure that viral accounts do not disproportionately influence the classifier, all numerical features were transformed using Min-Max normalization to the strict range $[0, 1]$. This approach was preferred over standardisation because social media counts, ratios, and growth rates are highly skewed, non-Gaussian, and possess a natural lower bound at zero that carries critical behavioural meaning. Furthermore, because our framework primarily leverages tree-based ensembles (such as Random Forest), the classifier remains structurally invariant to extreme

values; its recursive binary partitioning isolates heavy-tailed extremes into independent leaf nodes without skewing global decision boundaries. To prevent information leakage and guarantee model stability, the normalisation parameters were computed strictly on the training partition of each cross-validation fold, leaving binary and one-hot encoded features unchanged.

Several methods have been evaluated for feature selection, including Chi-square, Mutual Information, and Fisher's Score as filter methods, and Random Forest Importance as an embedded method. In addition, Shapley Additive exPlanations (SHAP) have been used as a model-agnostic interpretability technique to assess feature contributions. The analysis also distinguishes between features derived from the user's account and those from the content. Among these, Mutual Information (MI), Random Forest Importance (RFI) and SHAP values were retained for further analysis because they offer complementary perspectives on feature relevance, summarised below.

- **Mutual Information** is a non-parametric filter measure that captures any kind of statistical dependence between a feature and the target, including non-linear relationships. It is computed independently for each feature and is model-agnostic, which makes it a useful sanity check against model-specific biases.
- **Random Forest Importance** is an embedded measure that naturally accounts for feature interactions through the recursive splits of the tree ensemble. Because RFI is known to be biased towards features with high cardinality and many possible split points [60], we always interpret it in conjunction with MI and SHAP rather than in isolation, and we corroborate its rankings with model-agnostic measures.
- **SHAP values** provide a game-theoretic attribution of each feature's contribution to individual predictions, and aggregate naturally to a global ranking. Because our final model is a Random Forest, we use the *TreeSHAP* variant [61], which computes exact Shapley values for tree ensembles in polynomial time. We deliberately avoid KernelSHAP because it is an approximation that scales poorly with the number of features, and we avoid LIME because its locally linear surrogate is well suited for individual explanations but not for the global, cross-dataset comparison we pursue here. We also considered *permutation importance* as a model-agnostic alternative, but it is known to give unstable rankings in the presence of correlated features [62], which is a real concern in our case (e.g., *reputation* and *followers_friends_ratio* are very similar by construction).

A classification study has been carried out with the intention of comparing the accuracy obtained with each subset of features, as well as the computation time required for each run. The stopping criterion for the selection of the features was chosen to be that there is no improvement in two consecutive iterations of the maximum accuracy obtained in the previous iterations. We also experimented with longer patience values (3 and 4 iterations) and observed no qualitative change in the selected subsets, indicating that the criterion is conservative enough for our setting.

For classification model selection, a total of 15 different baselines have been compared. Different metrics have been evaluated in each of them, as well as their training time. In each model, cross-validation has been carried out with 10 partitions, and the average of the metrics obtained in each partition has been taken as the final value. All baselines were trained with their default hyperparameters as exposed by the PyCaret library, in order to keep the comparison fair across models and avoid favouring any particular family through more intensive tuning.

The evaluation metrics used throughout the experiment are the usual ones employed in classification systems, and are defined as follows:

$$\begin{aligned} Accuracy &= \frac{No.\ of\ correct\ predictions}{Total\ number\ of\ predictions} \\ &= \frac{TP + TN}{TP + TN + FP + FN} \end{aligned} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1\ Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{9}$$

where TP, TN, FP, and FN represent, respectively, the number of true positives, true negatives, false positives, and false negatives. In addition, we also use the Area Under the Curve, or AUC, since many of the problems related to bot detection are unbalanced. All three benchmark datasets used in this study are relatively balanced between bot and human classes (see Tables 5–7), so no resampling, class-weighting, or oversampling was applied; reporting both accuracy and AUC together protects against any residual imbalance effect on the interpretation of the results.

The experiments have been performed on a computer with the following components: AMD Ryzen 7 5800X (CPU), (AMD, Santa Clara, CA, USA) and Samsung SSD 980 PRO 1TB M.2 (DISK) (Samsung Electronics, Suwon, Republic of Korea).

4.3. Results

In this section, we present the outcomes and observations derived from the conducted experiments aimed at addressing the research questions outlined in the preceding sections.

In Figure 4, we can see the ranking of features in the three datasets using Mutual Information and Random Forest Importance. In addition, we can see where each feature comes from, in green the features coming from the account, and in blue the ones coming from the content. The two methods agree on which features are placed near the top, but their precise ordering can differ. This is expected: MI is a marginal, non-linear measure that scores each feature on its own dependence on the target, whereas RFI scores features by how much they reduce impurity once interactions with other features are accounted for. As a consequence, features that are individually informative but partially redundant with others tend to keep their rank in MI but drop in RFI, while features whose value emerges through interactions tend to gain rank in RFI. We treat such disagreements as informative rather than contradictory, and rely on the agreement with SHAP values as a tie-breaker. We also note that account-based features dominate the top of the rankings in TwiBot-20 more clearly than in the Cresci datasets, which is consistent with the greater diversity of bot types in TwiBot-20: with more heterogeneous bot families, individual content signatures become less discriminative on their own, while structural account properties (verification, reputation, growth rates) generalise better across bot types.

(**a**) Mutual Information Cresci-15

(**b**) R.F. Importance Cresci-15

(**c**) Mutual Information Cresci-17

(**d**) R.F. Importance Cresci-17

(**e**) Mutual Information TwiBot-20

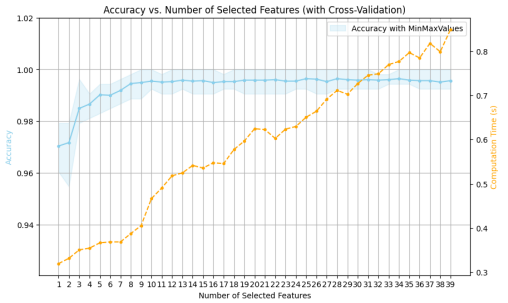
(**f**) R.F. Importance TwiBot-20

Figure 4. Feature ranking for all datasets.

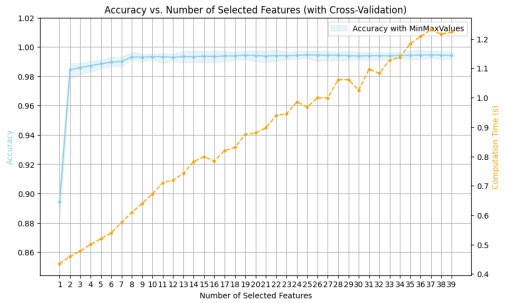
After analysing the features, Random Forest Importance was selected as the primary method for feature selection. This choice is motivated by several considerations. First, filtering methods do not account for correlations between features, which can lead to sub-optimal selections. In contrast, embedded methods like Random Forest not only consider feature interactions but also tend to offer greater accuracy, even if some generalisation

capability is sacrificed. For these reasons, embedded approaches are generally preferred over filtering and wrapper methods [63]. We acknowledge that this choice introduces a methodological dependency between feature selection and downstream classification (Random Forest is used in both stages), which may favour tree-based models in the model selection step. To partially mitigate this bias, we cross-check the resulting rankings against the model-agnostic SHAP and MI rankings reported in Figure 4, and observe that the top features identified by RFI also rank highly under the other two methods. Moreover, the final sets of features described in the following sections do not depend on this single measure: they combine Mutual Information, Random Forest importance and TreeSHAP on equal terms, so the generalisation framework that constitutes our main contribution is not tied to a tree-based importance.

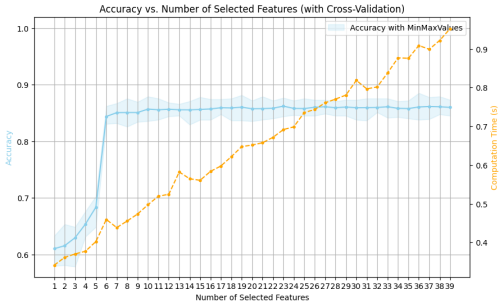
In Figure 5, we can see 40 runs with cross-validation of 10 of a Random Forest on each dataset, comparing the accuracy obtained with each subset of features as well as the computation time required for each run. Two observations follow from this figure. First, accuracy reaches a clear plateau after a relatively small number of features (around 6–10), beyond which adding more features yields only marginal gains while computation time grows roughly linearly. This indicates that a sizeable portion of the discriminative information is concentrated in a small subset of features, which is consistent with the bias–variance trade-off and supports the practical usefulness of feature selection: in deployment, a compact subset can deliver near-optimal accuracy at a fraction of the cost. Second, TwiBot-20 plateaus at a lower accuracy than the Cresci datasets, which we attribute to its greater bot diversity (it contains bots from politics, business, entertainment and sports domains, identified through crowdsourcing rather than a single capture campaign, and is therefore intrinsically harder).



(**a**) Cresci-15



(**b**) Cresci-17



(**c**) TwiBot-20

Figure 5. Results of classification depending on the number of features selected and the computation time using RF.

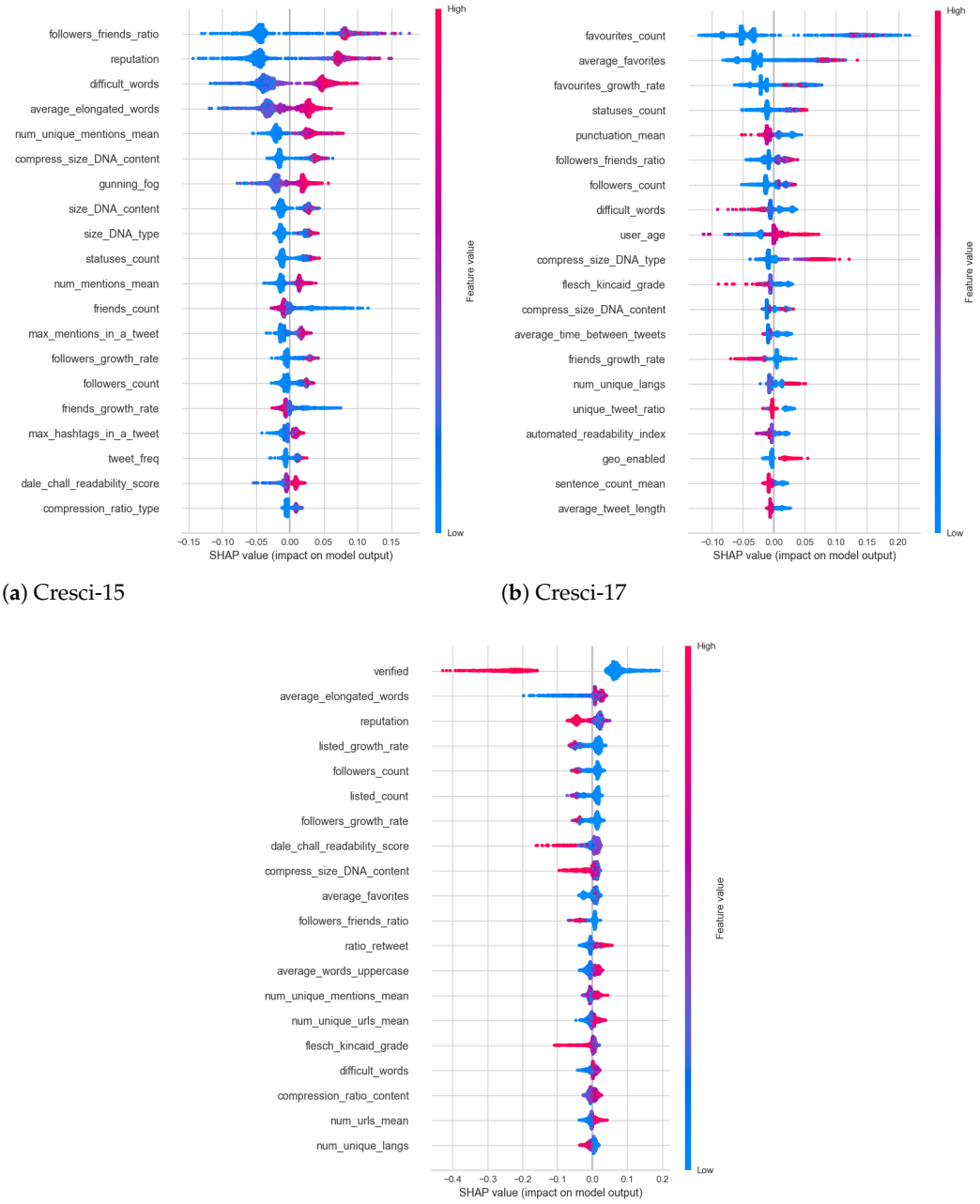
In Table 8, we can see the results of model selection in the Cresci-17 dataset. It should be noted that since the feature selection method is Random Forest Importance, it is expected that this same model will be among the first positions, since embedded methods

are dependent on the model in which they are embedded. The performance gap between the top tree-based models (Random Forest, LightGBM, Extra Trees, XGBoost) is below one percentage point in accuracy, which is well within the cross-validation standard deviation we observe, so we treat them as substantially equivalent in performance and choose Random Forest for its widespread availability and interpretability through TreeSHAP. The reported AUC of 0.00 for SVM with linear kernel and Ridge Classifier is not a true performance value: both models, in the PyCaret default configuration, do not expose probability estimates and were not calibrated, so their AUC is reported as 0.00 rather than computed from a valid score distribution. The very high accuracy on Cresci-17 is consistent with results reported in the literature for this dataset, and is not due to overfitting: we use 10-fold stratified cross-validation, accuracy drops to 85% on the harder TwiBot-20 dataset, and the held-out evaluation on TwiBot-22 reaches 81% under the same pipeline.

Table 8. Cresci-17 classification model selection (top 8 features). AUC reported as 0.00 for SVM-Linear and Ridge because these classifiers do not expose calibrated probabilities under the default PyCaret configuration; it is a reporting artefact, not a measure of poor ranking ability. Accuracy and F1 are valid for these rows. Bold indicates the best-performing model.

Model-Cresci-17	Acc.	AUC	Rec.	Prec.	F1	TT (Sec)
Random Forest Classifier	**99.43**	**99.97**	99.01	**98.65**	**98.83**	3.0560
Light Gradient Boosting Machine	99.39	**99.97**	98.89	98.62	98.75	2.0060
Extra Trees Classifier	99.37	99.94	**99.51**	97.94	98.72	1.8740
Extreme Gradient Boosting	99.35	**99.97**	98.93	98.42	98.67	1.5990
Gradient Boosting Classifier	99.29	99.92	98.56	98.53	98.54	8.7260
Ada Boost Classifier	99.16	99.91	98.23	98.32	98.27	2.8840
Decision Tree Classifier	98.77	98.30	97.41	97.50	97.45	1.5360
K Neighbors Classifier	98.24	99.07	95.48	97.21	96.33	1.4490
Linear Discriminant Analysis	96.91	99.14	91.24	95.77	93.42	1.4290
SVM-Linear-Kernel	96.65	00.00	92.19	93.93	92.98	1.4020
Ridge Classifier	96.27	00.00	88.85	95.43	91.98	1.3950
Logistic Regression	96.21	98.82	89.14	94.88	91.89	1.4930
Quadratic Discriminant Analysis	87.42	97.48	98.60	66.32	79.23	1.3580
Naive Bayes	86.23	92.20	98.93	64.79	78.09	1.3020
Dummy Classifier	75.82	50.00	00.00	00.00	00.00	1.1680

To further analyse feature importance, we visualise SHAP values in Figure 6, which illustrates the impact of each feature on the model's predictions at both global and local levels. The colour encodes the feature value (red for high, blue for low) and the horizontal axis is the SHAP value (positive contributions push the prediction towards the "bot" class, negative ones towards "human"). The pattern is consistent with domain intuition: low *followers_friends_ratio*, low *reputation*, high *average_elongated_words* count and a non-verified status all push predictions towards bot, while their opposites push towards human. We also observe that several top-ranked features are pairwise correlated by construction (e.g., *reputation* and *followers_friends_ratio*); TreeSHAP allocates contributions across correlated features following the tree path structure, so the importance of any single one may be slightly diluted with respect to a marginal measure. This is a known limitation of attribution under correlation, and is part of why we cross-check rankings with MI and RFI rather than relying on SHAP alone.



(**a**) Cresci-15 (**b**) Cresci-17

(**c**) Twibot-20

Figure 6. SHAP interpretability analysis for the Cresci-15, Cresci-17 and Twibot-20 datasets.

The primary objective of this study was to make a feature engineering process to evaluate the importance of the features in bot detection and improve the performance of the existing models. In order to compare our work with the previous works located in the literature, we have selected the studies that implement non-graph-based methods on the three datasets we worked on. In addition, we have chosen the three best baselines implemented in [16] by the authors, as in several cases they obtain better results than other works. In Table 9, the comparison of these methods with our proposal can be seen.

Table 9. Comparison between different non-graph-based methods from literature across datasets. The first three methods of each dataset are baselines implemented in [16]. Bold values indicate the best result for each metric.

Dataset	Method	Acc.	Prec.	Rec.	F1	AUC
C-15	Kouvela et al. [16,64]	97.80	99.50	96.80	98.20	-
	Lee et al. [16,44]	98.20	98.70	98.50	98.60	-
	LOBO [16,65]	98.40	98.50	99.10	98.80	-
	SCL [66]	98.00	-	-	98.00	-
	Fame for sale [49]	99.40	**99.70**	99.00	**99.40**	-
	OURS	**99.57**	99.35	**99.49**	**99.41**	**99.99**
C-17	Kouvela et al. [16,64]	98.40	99.20	99.00	99.10	-
	Lee et al. [16,44]	98.80	**99.60**	99.10	**99.30**	-
	LOBO [16,65]	96.60	99.30	96.10	97.70	-
	SCL [66]	98.00	-	-	98.00	-
	Bottrinet [67]	96.20	-	-	93.00	-
	OUProfiling [68]	98.10	-	-	97.60	-
	Ilias et al. [41]	99.06	98.00	**99.73**	98.86	99.74
	Deepsbd [69]	97.40	98.85	-	97.94	-
	OURS	**99.43**	98.65	99.00	98.83	**99.97**
T-20	Kouvela et al. [16,64]	84.00	79.30	**95.20**	86.50	-
	Lee et al. [16,44]	76.30	76.60	83.70	80.00	-
	LOBO [16,65]	75.70	74.80	87.80	80.80	-
	SCL [66]	82.00	-	-	86.00	-
	GA [70]	76.00	-	-	78.00	-
	GP [70]	76.00	-	-	78.00	-
	OURS	**85.43**	**83.20**	93.61	**87.93**	**92.62**

4.4. Hyperparameter Optimisation

While the fifteen baselines in Table 8 were compared under their PyCaret default settings to keep the model-family comparison fair, the selected Random Forest was subsequently optimised through a dedicated search over the following space: number of estimators {100, 200, 300, 500}, maximum depth {None, 10, 20, 30}, minimum samples to split {2, 5, 10}, minimum samples per leaf {1, 2, 4}, and maximum features per split {sqrt, log2, 0.5}. The search was conducted with a successive-halving strategy, which traverses the full grid while allocating progressively more training data to the more promising configurations, and the accuracy of the resulting configuration was then re-evaluated under the same 10-fold stratified cross-validation used throughout the paper. Table 10 reports, for each dataset, the accuracy of the default configuration, the accuracy of the tuned configuration, and the selected hyperparameters.

Table 10. Default versus tuned Random Forest accuracy (10-fold stratified cross-validation) and selected hyperparameters per dataset.

	Accuracy		Best Hyperparameters				
Dataset	Default	Tuned	n_{est}	Depth	Split	Leaf	Feat.
Cresci-15	99.57 (±0.29)	99.57 (±0.22)	100	None	5	1	sqrt
Cresci-17	99.43 (±0.11)	99.47 (±0.13)	500	20	2	1	0.5
TwiBot-20	85.43 (±1.19)	86.13 (±0.74)	200	30	2	4	0.5

4.5. Ablation Study

In this section, our objective is to determine the influence of different types of features on the classification system. To achieve this, we conducted an ablation study on our sets of characteristics. This involved systematically assessing the impact of using solely content-

based features, solely account-based features and a combined approach incorporating both types. It is important to note that, typically, ablation studies involve the addition or removal of different layers of data processing. In our case, as we examine the differences between sets of features, our approach to the ablation study focuses on understanding the final behaviour of the classifier based on which features and combinations achieve optimal performance. We also performed an ablation study using two distinct feature selection methods. We selected Mutual Information among filter methods, and Random Forest among embedded methods. In Figure 7, we present the results across different datasets and feature selection methods for the account features. In contrast, Figure 8 illustrates the results focused solely on content features.

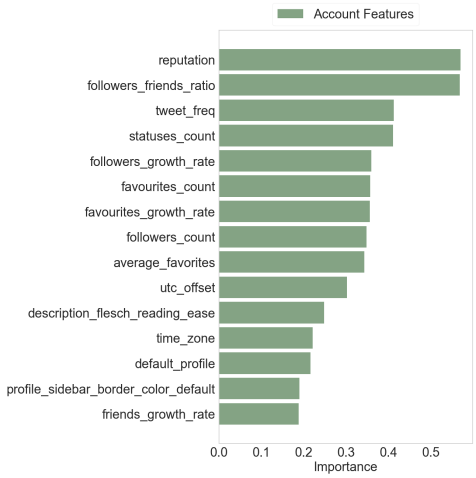
It is worth noting that, despite some ranking variations, both feature selection techniques consider the same characteristics. This alignment is essential for answering RQ1, which asks which features characterise a bot. Our findings suggest that, in terms of account-based features, those related to social interactions are paramount. Additionally, we observe the importance of temporal features like *tweet_freq* and *user_age*. In terms of content features, the style and complexity of writing emerge as the most influential characteristics for identifying bots. Notably, the features introduced by [41] prove to be particularly relevant in this context.

In determining the significance of feature types for bot categorisation on social media, Table 11 presents results for the best classification model when considering only account features, only content features, and a combination of both. The results represent the average outcomes across multiple executions. From the findings, it becomes evident that account features hold greater importance in the dataset and exhibit better capabilities for categorising bots. Notably, the most favourable result is associated with a combination of both account and content features.

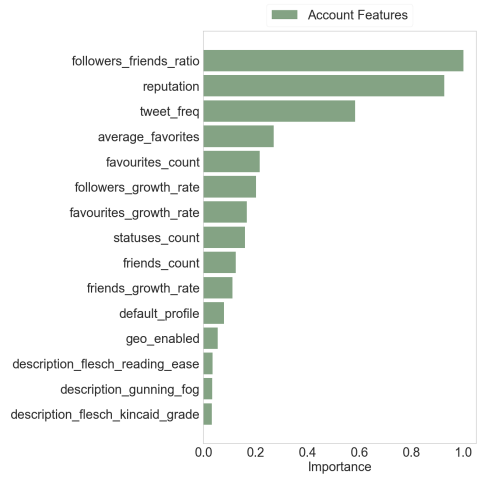
Table 11. Ablation study comparison in terms of accuracy. Bold values indicate the best accuracy per dataset.

Configuration-Dataset	C-15	C-17	T-20
Account Features	98.81	99.12	76.79
Content Features	98.65	94.14	68.27
Content + Account	**99.57**	**99.43**	**85.43**

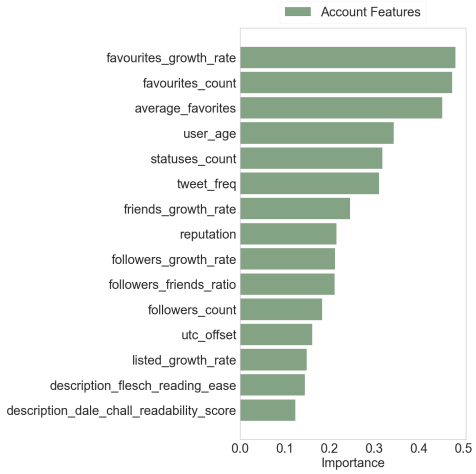
A few additional observations are worth making about Table 11. First, the gap between account-only and content-only features is dataset-dependent: it is negligible in Cresci-15 (a homogeneous dataset with a single bot type) and pronounced in Cresci-17 and TwiBot-20, where multiple bot families coexist. This is consistent with the intuition that content signatures generalise less well across bot types than structural account properties. Second, in every dataset, the combined configuration outperforms either source alone, which is the result we exploit in the rest of the paper. We do not report finer-grained removals (e.g., dropping only readability features, only colour-binning features, only DNA-based features) in this table because the broader question we wanted to answer here is the relative weight of the two information sources; finer-grained ablation can be derived directly from the rankings shown in Figures 4 and 6, where the importance of each individual feature is reported.



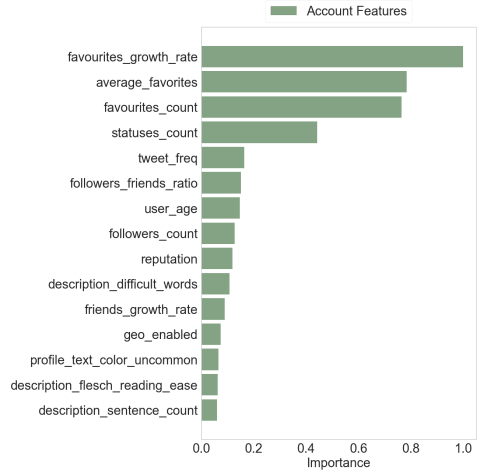
(**a**) Mutual Information Cresci-15.



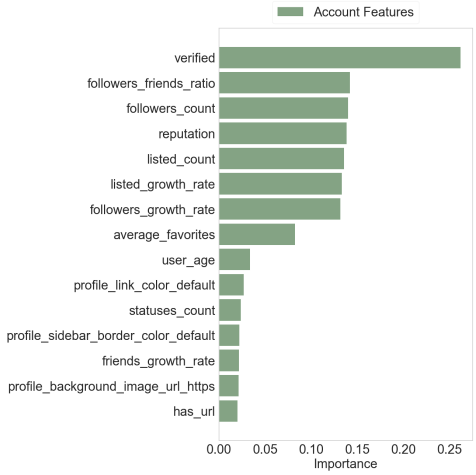
(**b**) R.F. Importance Cresci-15.



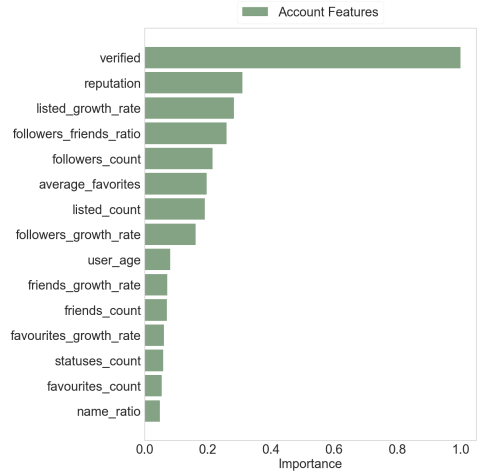
(**c**) Mutual Information Cresci-17.



(**d**) R.F. Importance Cresci-17.

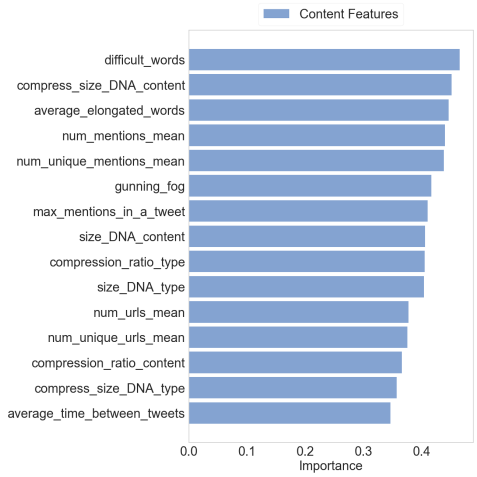


(**e**) Mutual Information Twibot-20.

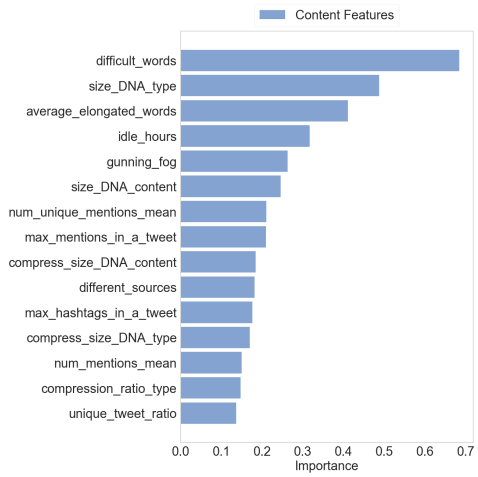


(**f**) R.F. Importance Twibot-20.

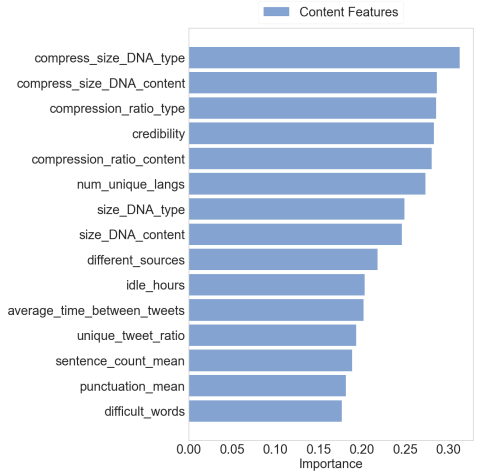
Figure 7. Significance of 15 most crucial account features across benchmark datasets employing various selection algorithms.



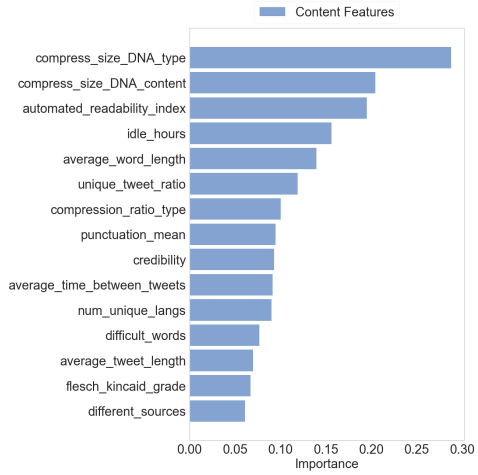
(**a**) Mutual Information Cresci-15.



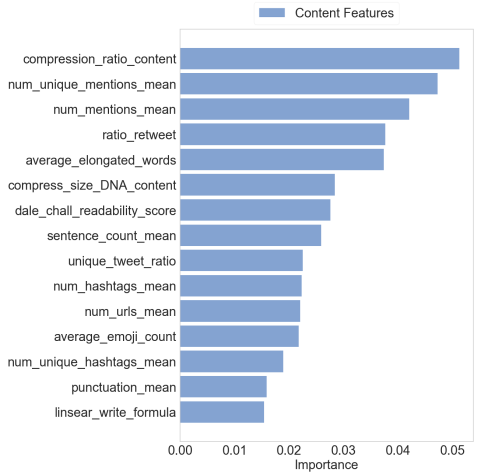
(**b**) R.F. Importance Cresci-15.



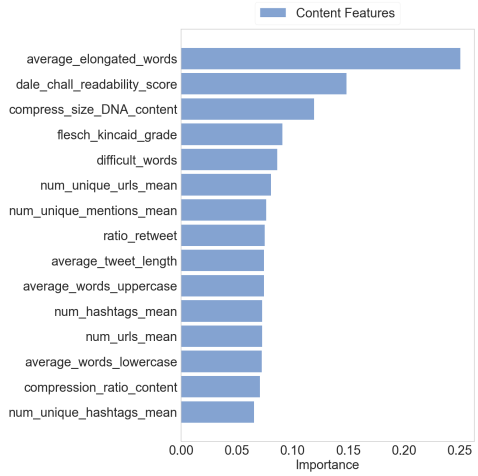
(**c**) Mutual Information Cresci-17.



(**d**) R.F. Importance Cresci-17.



(**e**) Mutual Information Twibot-20.



(**f**) R.F. Importance Twibot-20.

Figure 8. Significance of 15 most crucial content features across benchmark datasets employing various selection algorithms.

4.6. Robust Feature Set for Bot Detection

To identify features that generalise across different bot detection datasets, we propose a multi-dataset feature selection approach that combines three importance metrics, eval-

uated across three distinct datasets. For each dataset D_i ($i \in \{1, 2, 3\}$), we compute three feature importance rankings: R^i_{MI} (Mutual Information), R^i_{RF} (Random Forest Importance), and R^i_{SHAP} (SHAP Values). Each ranking is normalised so that the importance scores are scaled between 0 and 1 within each method. We then compute a combined importance score for each feature f in dataset D_i:

$$S_i(f) = \frac{1}{3}\Big(N(R^i_{\text{MI}}(f)) + N(R^i_{\text{RF}}(f)) + N(R^i_{\text{SHAP}}(f))\Big) \tag{10}$$

where $N(\cdot)$ denotes the normalisation function that scales the raw importance scores to the range [0, 1]. We adopt a simple unweighted average of the three normalised scores for two reasons. First, the three methods have no a priori reason to be weighted differently in our setting: each captures a distinct aspect of relevance (non-linear marginal dependence, interaction-based importance, and game-theoretic attribution) and we do not have ground-truth importances against which a weighting could be learned. Second, the unweighted average is the most conservative aggregation: it does not bias the combined score towards any single method and is robust to small changes in the underlying rankings. We verified that the resulting universal, common and specific sets are stable under small perturbations of the weights.

The top features F_i for each dataset are selected as:

$$F_i = \{f \mid S_i(f) > \tau_i\} \tag{11}$$

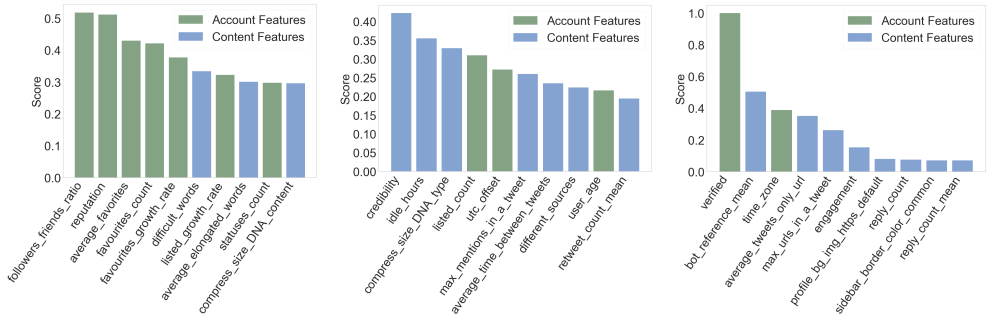
where τ_i is a threshold that retains the top third of features. The top-third cut-off is a deliberate compromise between coverage and selectivity: a stricter threshold (top quartile or top tenth) yields very small universal/common sets that are unstable across datasets, while a more permissive one (top half) dilutes the universal set with features that are clearly more dataset-specific. We confirmed empirically that the universal and common sets reported below are stable under alternative thresholds in the range top-25% to top-40%, with only borderline features moving between the common and specific categories.

This approach reduces noise and improves the quality of the feature set while maintaining a balance between feature importance and dataset representation. After that, we divide each feature into different sets:

1. Universal set: $F_{\text{universal}} = F_1 \cap F_2 \cap F_3$ (12)
2. Common set: $F_{\text{common}} = (F_1 \cap F_2) \cup (F_1 \cap F_3) \cup (F_2 \cap F_3)$ (13)
3. Specific set: $F^i_{\text{specific}} = F_i \setminus (F_{\text{common}} \cup F_{\text{universal}})$ (14)

Universal features are those that appear in the top third of all datasets, common features are those that appear in the top third of at least two datasets, and specific features appear in the top third of only one dataset. The idea behind this segmentation is to identify which features might have more predictive power when it comes to identifying bots, regardless of the dataset we are working with, and also to identify those features that are relevant enough in a single dataset to outperform the most common features.

The rankings of the three sets of features and the origin of these features can be seen in Figure 9. In addition, a global ranking of the union of the three sets can be seen in Figure 10.



(**a**) Universal Features (**b**) Common Features (**c**) Specific Features

Figure 9. Feature importance rankings for different feature sets.

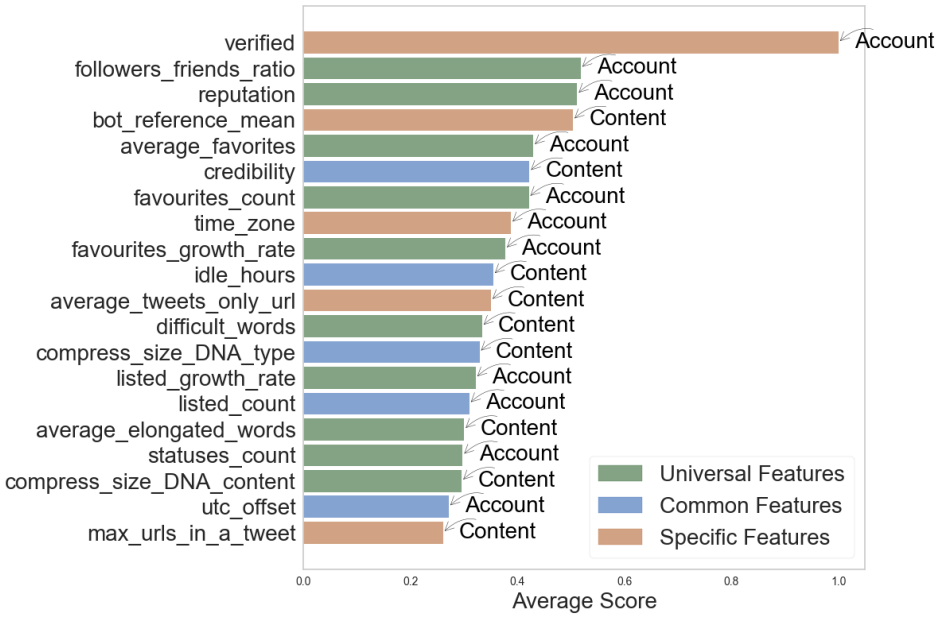


Figure 10. Top 20 most important features across sets.

In order to evaluate the effectiveness of the selected features and sets, several experiments were carried out on a dataset that, as far as we know, is one of the most recent and much larger, the TwiBot-22 [16] dataset. Importantly, the universal, common and specific partition was derived exclusively from the three earlier datasets (Cresci-15, Cresci-17 and TwiBot-20); TwiBot-22 was held out entirely from the selection process and used only to evaluate generalisation. This ensures that no information about TwiBot-22 leaks into the definition of the universal set, and that the reported numbers are a fair test on an unseen dataset. Several experiments and comparisons were carried out, using Random Forest as the base model, first training a model with all the features, then training different models with the feature sets and with the 20 most important features of the union of the sets. The results are shown in Figure 11 and Table 12 and, as cross-validation with 10 partitions has been used, the results are the average of the data obtained.

Table 12. Classification performance metrics using different feature subsets. *UF Count* refers to the number of usable features processed in the dataset from each subset.

Feature Set	Accuracy	AUC	Recall	Precision	F1	UF Count	Time
All Features	81.05 (±0.25)	88.68 (±0.17)	80.78 (±0.43)	81.23 (±0.38)	81.00 (±0.24)	136	79.15 s
Top 20 Features	78.69 (±0.33)	85.61 (±0.26)	79.03 (±0.38)	78.50 (±0.43)	78.76 (±0.31)	18	29.25 s
Universal	77.89 (±0.29)	84.57 (±0.23)	78.05 (±0.50)	77.80 (±0.37)	77.92 (±0.30)	10	25.60 s
Common	76.17 (±0.28)	82.59 (±0.29)	75.30 (±0.67)	76.64 (±0.46)	75.96 (±0.31)	9	22.91 s
Specific	69.55 (±0.35)	76.22 (±0.26)	69.71 (±0.68)	69.49 (±0.38)	69.60 (±0.41)	7	9.96 s

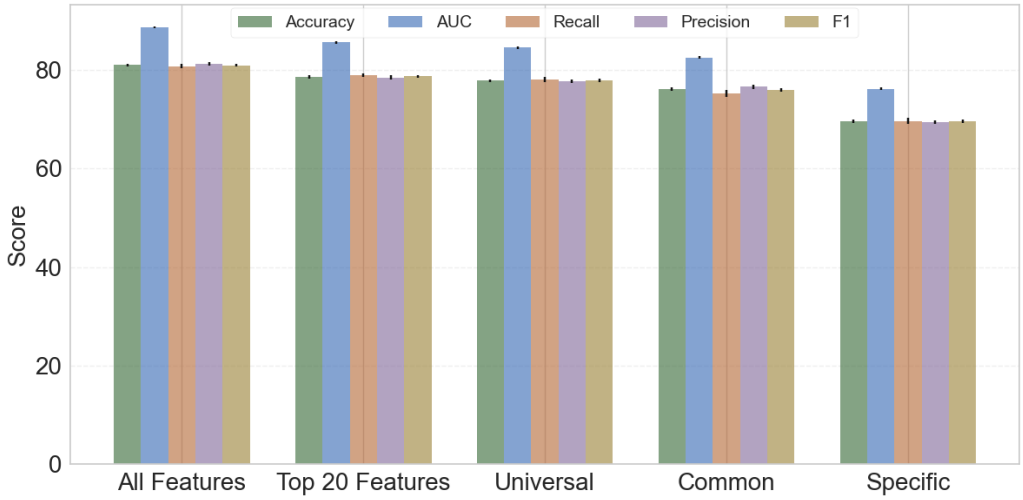


Figure 11. Performance comparison by feature set.

As expected, using the full set of features yields the best overall performance across all metrics, with the highest accuracy (81.05), AUC (88.68), and F1-score (81.00), but it also requires the longest training time and the largest number of features (136). Among the three defined tiers, the universal feature set performs the best, achieving reasonably high scores with a significantly smaller number of features and under a third of the training time. This suggests that features consistently important across all datasets do retain strong predictive power, even when applied to a new and challenging dataset like TwiBot-22. The common feature set has slightly worse results than the universal set, suggesting that features important in only two datasets may not generalise as effectively. In contrast, the specific feature set performs considerably worse across all metrics, indicating that features relevant in only one dataset may lack generalisability or offer limited complementary information for bot detection.

Interestingly, the top 20 features, selected from the union of all feature sets based on their aggregated importance scores, provide a solid balance. This subset achieves competitive results with the universal set, slightly outperforming it in most metrics. The reason is that the Top 20 set, although larger than the universal set, draws features from all three categories (universal, common, and dataset-specific) according to their aggregated score. This brings in features that are highly relevant in one or two datasets and happen to also be informative in TwiBot-22, complementing the universal signals. The universal set, by construction, only retains features that are simultaneously in the top third of all three source datasets, which is a strict condition and discards informative features that just miss the cut-off in one of them. The fact that adding these complementary features only yields a modest gain (about 1 percentage point in accuracy) is itself informative: most of the generalisable signal is already captured by the universal set, and the marginal improvement comes at the cost of 8 additional features and a slightly longer training time. The differences between universal, top-20 and common fall within roughly two standard deviations, so we interpret them as small but consistent. The specific set, on the other hand, sits well below the rest in every metric and across standard deviations, which we interpret as evidence that purely dataset-specific features carry limited cross-dataset value.

5. Discussion

Although deep learning has led to significant progress in areas such as computer vision, natural language processing (NLP) and pattern recognition, it frequently compromises interpretability. By contrast, traditional machine learning models, such as decision trees or linear classifiers, offer greater transparency, making it easier to understand how

individual features influence predictions. This interpretability is particularly valuable for tasks that require insight into model behaviour, such as debugging or understanding feature relevance.

In this study, we opted for a feature-driven approach using classical machine learning techniques. This approach not only retained interpretability but also achieved competitive and, in some cases, superior performance compared to more complex, state-of-the-art methods that rely on both user metadata and content. This is especially relevant in the practical approach because, even if you do not want to follow a detection algorithm based on machine learning, this study can serve as a basis to identify which features you should pay special attention to when facing the task of identifying bots.

Compared to deep-learning-based pipelines, our approach has three practical advantages. In terms of computational cost, a Random Forest with a few hundred trees on around 136 features trains in tens of seconds on a single CPU workstation (see Table 12), with a peak memory footprint below half a gigabyte and a serialised model of at most a few tens of megabytes. At inference time, classifying an account takes on the order of a tenth of a millisecond, so the model can score large volumes of accounts in real time, without any GPU. By contrast, transformer or graph-neural-network-based bot detectors typically need GPU resources and orders of magnitude more training time and memory, so the small differences in detection accuracy reported in Tables 9 and 13 are obtained at a fraction of the computational cost. Regarding feature availability, our approach depends only on account metadata and short content statistics, both of which can be obtained for any user with at least a handful of posts; graph-based approaches additionally require access to the follower/friend network, which is increasingly restricted under the current 𝕏 API policy. Finally, a feature-based RF model can be packaged as a self-contained scikit-learn artefact, audited feature by feature, and updated incrementally, whereas a black-box deep model is harder to audit and redeploy, which can be a problem in moderation contexts where explainability is required.

In addition to explainability, the operational viability of a detection framework heavily depends on its generalizability across differing digital ecosystems. While this study is empirically grounded in datasets from 𝕏, the underlying architecture is engineered around principles of platform-agnostic universality. We replaced complex network topology and platform-specific designs, with simple, universal features, so the framework remains highly portable. Extending this methodology to alternative platforms such as Facebook, TikTok, or Instagram does not require a structural change, but rather a systematic two-tiered adaptation:

- Attribute mapping: Aligning functionally equivalent core metrics across interfaces. For instance, the propagation mechanic of a "Retweet" on 𝕏 maps directly to "Shares" on Facebook or TikTok, while "Followers" translate semantically to "Friends" or "Subscribers" depending on the target network's social paradigm.
- Feature adaptation: Replacing highly dataset-specific variables of 𝕏 with native metadata that captures analogous behavioural dimensions, such as average video duration on TikTok or account classification (personal vs. professional) on Instagram.

With this in mind, we now turn to the research questions posed at the beginning of the study.

RQ. 1. *What features define a social bot?*

To answer this question, we will look at Figures 9 and 10. The first thing to note is that the top positions are occupied by features related to social interactions. Among them, we can highlight *reputation* and *followers_friends_ratio*, which are very similar features. The importance of the features related to favourites is very remarkable, as well as those related to lists. Next, we can see that the proposed measure *credibility*, also based on

interactions, is in sixth place in the overall ranking and in first place in the set of common features. With these observations in mind, we can intuit that it is the same people within the social network who, with their activity and interactions, give us the most important clues to differentiate an automatic account from a non-automatic one. Figure 12a shows how automatic accounts tend to have low values of reputation, as opposed to genuine accounts that are more evenly distributed across their values.

One noticeable observation is that temporal features are fairly well represented at the top of importance. Features such as *idle_hours*, *average_time_between_tweets* and *user_age* seem to have some predictive power and may be an indication that automatic accounts follow temporal patterns. The *user_age* is a particularly interesting feature. In our study, *user_age* signifies the duration an account has been active, measured from its creation to the moment the dataset is formed. Based on that, similar user ages imply that the respective accounts were created around the same time. This feature, a priori not very relevant, provides us with more information than it seems. As social networks have grown, the interest in creating bots has increased. As a consequence, at the beginning there were far fewer bots than before, so it is more likely that an account with a lot of age is genuine. Looking at Figure 12c, we can see how the distributions of user age in bots and genuine accounts support our theory.

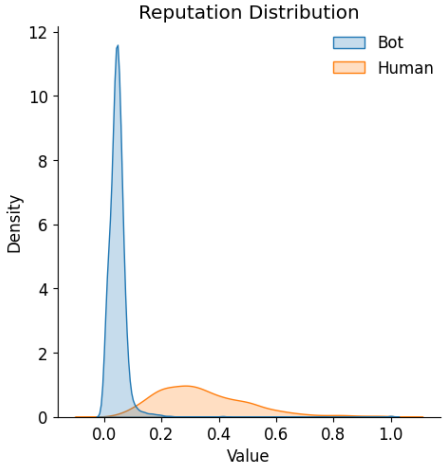
Other features classified as temporal that appear in the rankings are those given in [55], referred to as DNA in Table 2. These measures are related to the activity patterns of the users and can give us clues that there is some automation and sequences of activities that are repeated in certain accounts. This can be seen in Figure 12d, which is especially interesting as it shows that when the number of tweets is low the difference between the size of the compressed DNA of a bot and a genuine account is very small (peaks at 50) but, as can be seen, when the number of tweets of both classes increases this difference also increases (peaks at 120–140). This tells us that this feature set is interesting to use when we are dealing with a large volume of tweets per user; in practical terms, DNA-based features become reliably discriminative when an account has accumulated on the order of 100 or more tweets, and should be expected to be weak for newly created or low-activity accounts.

We can see that in the rankings, there are features related to stylometry and readability. In particular we can see how *difficult_words* and *average_elongated_words* appear. An example of this can be seen in Figure 12b, where we can see how the bots of the Cresci-15 dataset tend to use less elongated words on average than real users. From a linguistic standpoint, elongated words (such as “soooo” or “yessss”) are typical of informal human writing; automated accounts, whether rule-based or LLM-generated, tend to favour standard orthography.

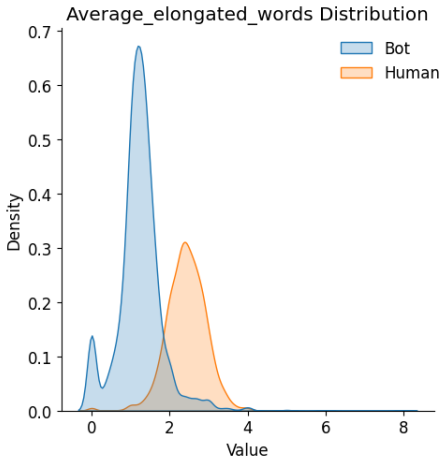
It is important to acknowledge that, looking at the per-feature distributions in Figure 12, the overlap between bot and human classes is substantial in every panel. None of these features is, in isolation, a clean separator: a single threshold would misclassify a non-trivial fraction of users no matter where it is placed. We do not interpret the rankings of Figures 4, 6, 9 and 10 as a claim that any single feature is sufficient. Rather, the rankings tell us which features carry useful signal once combined by the classifier, which is the regime that actually matters in deployment. The high accuracy we obtain in spite of the univariate overlap is precisely the contribution of the multivariate model: the Random Forest exploits feature interactions (low reputation short user age and few elongated words, for example) to push samples that look ambiguous on any single axis into the correct class. In practice, the remaining misclassifications are accounts that look human-like across many features at once.

Finally, we observe the most evident features, those that one might intuitively expect to have a significant impact from the beginning. Among these, the *verified* status

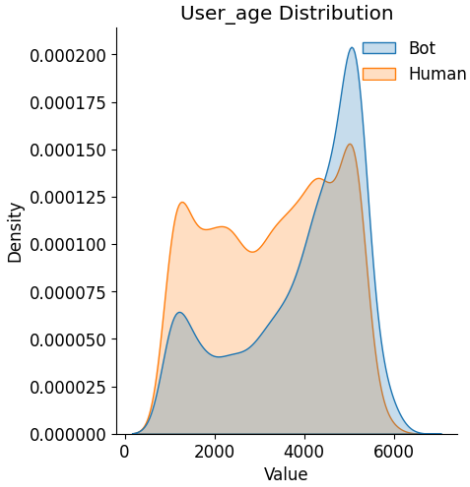
stands out as the most predictive feature across all methods. Another important variable is *bot_reference_mean*, which reflects how often the word 'bot' appears in an account's content and can serve as a strong indicator of automation. In addition, features related to mentions, such as *average_tweets_only_url* and *max_urls_in_a_tweet*, also prove to be relevant, suggesting patterns typically associated with automated behaviour.



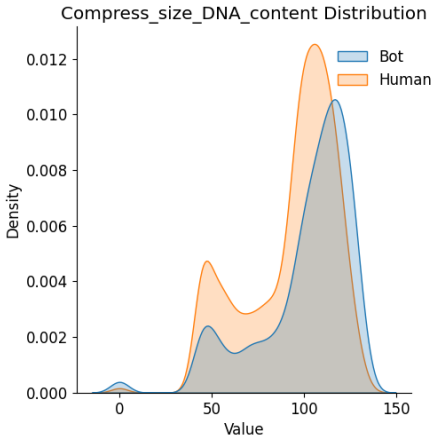
(**a**) Reputation in Cresci-15



(**b**) Average elongated words in Cresci-15



(**c**) User age in Twibot-20



(**d**) Compress size DNA content in Twibot-20

Figure 12. Feature distribution across the datasets.

RQ. 2. *Which source of features holds greater importance in social bot detection, account-based or content-based features?*

To answer this question, we must look at Figure 4 and the ablation study performed in the previous section. In Figure 4, we can see how in the three datasets the first positions of the top are occupied by the features coming from the account, this agrees well with what is observed in Table 11 where we can see that only using features coming from the user's account we reach a higher accuracy than using features coming from the content. With this we could consider the question answered if it were not for a nuance, taking a closer look at Table 11 we can see that in the Cresci-15 dataset the difference between the accuracy obtained with the two sources of features is very small, but this difference is accentuated in the other two datasets. If we remember the Cresci-15 dataset consists of only one type of bot, while the other two contain more variety. So we can say that in general the features coming from the account have more relevance when it comes to giving a classification, but it

is possible that the performance of the features coming from the content varies according to the type of bot we are trying to detect.

Finally to solve this question is worth mentioning that in today's era, with the presence of generative AIs, it has become effortless to generate and illicitly acquire personal information, particularly related to account features like profile images or backgrounds. This ease of access empowers bots to project greater confidence by leveraging these features. Consequently, this underscores the essential need for both content and account features to collaborate, emphasizing the importance of a united approach to address this challenge.

RQ. 3. *Can a social bot be identified based on user-profile features? Are they enough?*

To answer this question, we will rely on the results shown in Table 13, where we have collected some of the most recent studies that use graphs as well as other techniques. As can be seen, in the cases of the Cresci-15 and TwiBot-22 datasets, our proposal is still superior to those presented. On the other hand, in the TwiBot-20 dataset, our proposal lags behind the best graph-based methods, such as BIC, in terms of accuracy and F1, although it remains competitive. The reason is that TwiBot-20 combines a highly heterogeneous bot population with a dense interaction graph, precisely the setting where network structure is most informative, models like BIC exploit the follower/friend connections that our method deliberately does not use. This means there is information in the network that account- and content-based features cannot recover, and if such a network is available, it could be worth using for detection. This does not imply that methods without networks are uncompetitive: in fact, they achieve results close to, and in some cases above, network-based ones, and they remain the preferred option precisely when the network is not usable, either because users are unwilling to expose all their data, use the platform only to promote a personal project, or simply have not written enough posts for many features to be relevant. Our approach performs best when the discriminative signal lies in account metadata and activity patterns, and is at its weakest, relative to graph models, only when a dense and reliable graph is available for a diverse bot population, in those cases, account-based detection is best seen as complementary to graph-based techniques rather than a replacement.

It is critical to address the evolving threat landscape posed by advanced Large Language Models (LLMs) like GPT-4 or Claude, which possess an unprecedented capability to replicate human linguistic nuances. While modern LLMs can successfully obfuscate bot identities by generating flawless, human-like text, our framework mitigates this challenge by architecturally shifting the analytical focus from content to structure. Our findings demonstrate that account-level metadata—such as account age, reputation metrics, and follower-to-friend ratios—constitute foundational structural signals that remain highly discriminative and inherently resilient against generative AI spoofing. Moreover, through the integration of "Social Fingerprinting" (DNA) features, our model analyses sequential and temporal activity patterns. Although an LLM-driven agent can synthesise natural prose, its underlying behavioural rhythm—specifically the timing, burstiness, and execution frequency of its actions—consistently exposes automated orchestration when contrasted with the erratic, non-deterministic habits of genuine users. We acknowledge as a limitation that benchmark datasets, including TwiBot-22, were collected prior to the widespread deployment of commercial LLMs and thus do not explicitly contain GPT-4 or Claude-driven agents. However, TwiBot-22 represents the exact evolutionary tipping point where traditional text-based detectors began to fail due to early sophisticated language models. Therefore, our competitive performance on TwiBot-22 empirically validates the core thesis of this work: as text generation becomes indistinguishable from humans, relying on account metadata and temporal DNA fingerprints is not only robust, but represents the most viable path forward for future-proof bot detection.

Table 13. Comparison between different graph-based baselines from literature. Bold values indicate the best result for each metric within a dataset.

Method	Cresci-15		TwiBot-20		TwiBot-22	
	Acc	F1	Acc	F1	Acc	F1
CGNN [71]	99.13 (±0.21)	99.33 (±0.16)	86.37 (±0.42)	88.14 (±0.37)	-	-
SAGE + CACL [72]	97.65 (±0.00)	98.12 (±0.00)	83.60 (±0.00)	86.53 (±0.00)	75.38 (±0.00)	47.45 (±0.00)
HGT + CACL [72]	95.88 (±0.00)	96.74 (±0.00)	85.12 (±0.00)	87.28 (±0.00)	75.07 (±0.00)	48.27 (±0.00)
LGB (GCN) [73]	-	-	85.09 (±0.51)	87.44 (±0.31)	80.42 (±0.05)	**81.31 (±0.08)**
BIC [74]	98.35 (±0.24)	98.71 (±0.18)	**87.61 (±0.21)**	**89.13 (±0.15)**	-	-
BotDGT [75]	-	-	87.25 (±0.51)	88.87 (±0.55)	79.33 (±0.22)	58.15 (±0.74)
OURS	**99.57 (±0.12)**	**99.41 (±0.15)**	85.43 (±0.21)	87.93 (±0.23)	**81.05 (±0.25)**	81.00 (±0.24)

6. Conclusions

In this paper, we emphasise the necessity of using a combination of account- and content-based features for effective bot detection. By creating a large set of features composed of literature features and new features proposed by us, and by applying a feature selection and combination process, we managed to surpass state-of-the-art results across three distinct benchmark datasets in several metrics using a random forest algorithm. We introduced two new features (profile colour personalisation and general-purpose credibility/engagement measures) and showed that they consistently appear among the most predictive features under three complementary importance criteria (Mutual Information, Random Forest Importance and TreeSHAP). We have provided detailed information on how to create these features using feature engineering techniques to support other researchers in incorporating them into their work.

Our findings highlighted the significance of defining bots through a fusion of account-based and content-based features. We observed that the importance of feature types varied based on the selection method employed, with account-based features proving more conducive to accurate classification. Furthermore, in distinguishing between different types of bots, we demonstrated the varying degrees of importance associated with different characteristics.

In addition, we have proposed a categorisation of features into *universal*, *common*, and *dataset-specific* sets, based on a multi-perspective importance score, and validated this categorisation on the more recent and larger TwiBot-22 dataset (which was held out entirely during the construction of the sets). The universal set, made up of only ten features, recovers most of the predictive performance of the full 136-feature pipeline at a fraction of the cost, providing a compact and transparent toolbox for practitioners. It is worth noting that in the era of generative AI, it has become easier to generate fake content or enhance fake account features, such as profile pictures. Therefore, systems that incorporate a variety of features, as proposed in our research, prove valuable in mitigating the proliferation of bot accounts.

As part of our future work, we aim to enhance our system by incorporating graph-based methods to capture the entirety of account interactions within their environments, addressing the gap with the best graph-based baselines observed on TwiBot-20. The inclusion of semantic information from content, along with exploring the impact of word embeddings, would enable the creation of bot prototypes. Additionally, we plan to incorporate user biographies and descriptions into the characteristic sets, further enriching the features for more nuanced bot detection. A particularly relevant direction is the evaluation of our universal feature set against LLM-powered bots, which are not represented in the benchmark datasets used here and are likely to require dedicated features.

Author Contributions: Conceptualization, S.L.-J., J.A.D.-G., M.D.R. and M.J.M.-B.; methodology, S.L.-J., J.A.D.-G., M.D.R. and M.J.M.-B.; software, S.L.-J.; validation, J.A.D.-G., M.D.R. and M.J.M.-B.; formal analysis, M.D.R. and M.J.M.-B.; investigation, S.L.-J. and J.A.D.-G.; resources, M.D.R. and M.J.M.-B.; data curation, S.L.-J.; writing—original draft preparation, S.L.-J. and J.A.D.-G.; writing—review and editing, M.D.R. and M.J.M.-B.; visualization, S.L.-J.; supervision, M.D.R. and M.J.M.-B.; project administration, M.D.R. and M.J.M.-B.; funding acquisition, M.D.R. and M.J.M.-B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the European Union and the Spanish Ministry of Science, Innovation, and Universities.

Data Availability Statement: Most of the datasets supporting the findings of this study are openly available in the Botometer repository at https://botometer.osome.iu.edu/bot-repository/datasets.html (accessed on 15 March 2025). Some of them, such as TwiBot-20 and TwiBot-22, are available on request from the corresponding author.

Acknowledgments: The research reported in this paper was supported by the European Union NextGenerationEU/PRTR, and FederaTrans project: Grant PID2024-158373OB-I00 funded by MICIU/AEI/10.13039/501100011033 and by ERDF/EU. Finally, the research reported in this paper is also funded by the European Union (BAG-INTEL project, grant agreement no. 101121309 and CUSTOMAI project, grant agreement no. 101226029).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Almars, A.M.; Atlam, E.S.; Noor, T.H.; ELmarhomy, G.; Alagamy, R.; Gad, I. Users opinion and emotion understanding in social media regarding COVID-19 vaccine. *Computing* **2022**, *104*, 1481–1496. [CrossRef]
2. Linvill, D.L.; Warren, P.L. Troll factories: Manufacturing specialized disinformation on Twitter. *Political Commun.* **2020**, *37*, 447–467. [CrossRef]
3. Nisbet, E.C.; Mortenson, C.; Li, Q. The presumed influence of election misinformation on others reduces our own satisfaction with democracy. *Harv. Kennedy Sch. Misinformation Rev.* **2021**, *1*. [CrossRef]
4. Kennedy, I.; Wack, M.; Beers, A.; Schafer, J.S.; Garcia-Camargo, I.; Spiro, E.S.; Starbird, K. Repeat spreaders and election delegitimization: A comprehensive dataset of misinformation tweets from the 2020 US election. *J. Quant. Descr. Digit. Media* **2022**, *2*, 1–49. [CrossRef]
5. Freelon, D.; Bossetta, M.; Wells, C.; Lukito, J.; Xia, Y.; Adams, K. Black trolls matter: Racial and ideological asymmetries in social media disinformation. *Soc. Sci. Comput. Rev.* **2022**, *40*, 560–578.
6. Shao, C.; Ciampaglia, G.L.; Varol, O.; Flammini, A.; Menczer, F. The spread of fake news by social bots. *arXiv* **2017**, arXiv:1707.07592.
7. Deseriis, M. Hacktivism: On the use of botnets in cyberattacks. *Theory, Cult. Soc.* **2017**, *34*, 131–152.
8. Hammi, B.; Zeadally, S.; Khatoun, R. An empirical investigation of botnet as a service for cyberattacks. *Trans. Emerg. Telecommun. Technol.* **2019**, *30*, e3537.
9. Hajli, N.; Saeed, U.; Tajvidi, M.; Shirazi, F. Social bots and the spread of disinformation in social media: The challenges of artificial intelligence. *Br. J. Manag.* **2022**, *33*, 1238–1253.
10. Miller, S.; Busby-Earle, C. The role of machine learning in botnet detection. In *Proceedings of the 2016 11th International Conference for Internet Technology and Secured Transactions (ICITST)*; IEEE: New York, NY, USA, 2016; pp. 359–364.
11. Nguyen, H.D.; Nguyen, D.Q.; Nguyen, C.D.; To, P.T.; Nguyen, D.H.; Nguyen-Gia, H.; Tran, L.H.; Tran, A.Q.; Dang-Hieu, A.; Nguyen-Duc, A.; et al. Supervised learning models for social bot detection: Literature review and benchmark. *Expert Syst. Appl.* **2023**, *238*, 122217. [CrossRef]
12. Gorwa, R.; Guilbeault, D. Unpacking the social media bot: A typology to guide research and policy. *Policy Internet* **2020**, *12*, 225–248.
13. Fan, R.; Talavera, O.; Tran, V. Social media bots and stock markets. *Eur. Financ. Manag.* **2020**, *26*, 753–777.
14. Weng, Z.; Lin, A. Public opinion manipulation on social media: Social network analysis of twitter bots during the COVID-19 pandemic. *Int. J. Environ. Res. Public Health* **2022**, *19*, 16376. [CrossRef] [PubMed]
15. Diaz-Garcia, J.A.; Ruiz, M.D.; Martin-Bautista, M.J. NOFACE: A new framework for irrelevant content filtering in social media according to credibility and expertise. *Expert Syst. Appl.* **2022**, *208*, 118063. [CrossRef]
16. Feng, S.; Tan, Z.; Wan, H.; Wang, N.; Chen, Z.; Zhang, B.; Zheng, Q.; Zhang, W.; Lei, Z.; Yang, S.; et al. TwiBot-22: Towards graph-based Twitter bot detection. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 35254–35269. [CrossRef]

17. Pastor-Galindo, J.; Marmol, F.G.; Pérez, G.M. Profiling users and bots in Twitter through social media analysis. *Inf. Sci.* **2022**, *613*, 161–183. [CrossRef]
18. Abokhodair, N.; Yoo, D.; McDonald, D.W. Dissecting a social botnet: Growth, content and influence in Twitter. In *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing*; ACM: New York, NY, USA, 2015; pp. 839–851.
19. Cresci, S. A decade of social bot detection. *Commun. ACM* **2020**, *63*, 72–83. [CrossRef]
20. Morstatter, F.; Wu, L.; Nazer, T.H.; Carley, K.M.; Liu, H. A new approach to bot detection: Striking the balance between precision and recall. In *Proceedings of the 2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*; IEEE: New York, NY, USA, 2016; pp. 533–540.
21. 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*; PKP Publishing Services: Burnaby, BC, Canada, 2020; Volume 34, pp. 1096–1103.
22. Assenmacher, D.; Clever, L.; Frischlich, L.; Quandt, T.; Trautmann, H.; Grimme, C. Demystifying social bots: On the intelligence of automated social media actors. *Soc. Media+ Soc.* **2020**, *6*, 2056305120939264. [CrossRef]
23. Ferrara, E.; Varol, O.; Davis, C.; Menczer, F.; Flammini, A. The rise of social bots. *Commun. ACM* **2016**, *59*, 96–104. [CrossRef]
24. Lopez-Joya, S.; Diaz-Garcia, J.A.; Ruiz, M.D.; Martin-Bautista, M.J. Bot Detection in Twitter: An Overview. In *Proceedings of the International Conference on Flexible Query Answering Systems*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 131–144.
25. Dontu, S.; Vallabhaneni, R.; Addula, S.R.; Pareek, P.K.; Balassem, Z.A. Cybersecurity framework development for BoTNet attack detection using issoa based attention recurrent autoencoder. In *Proceedings of the 2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS)*; IEEE: New York, NY, USA, 2024; pp. 1–7.
26. Hayawi, K.; Mathew, S.; Venugopal, N.; Masud, M.M.; Ho, P.H. DeeProBot: A hybrid deep neural network model for social bot detection based on user profile data. *Soc. Netw. Anal. Min.* **2022**, *12*, 43. [CrossRef] [PubMed]
27. 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*; ACM: New York, NY, USA, 2019; pp. 183–192.
28. Heidari, M.; Jones, J.H.; Uzuner, O. Deep contextualized word embedding for text-based online user profiling to detect social bots on twitter. In *Proceedings of the 2020 International Conference on Data Mining Workshops (ICDMW)*; IEEE: New York, NY, USA, 2020; pp. 480–487.
29. Pennington, J.; Socher, R.; Manning, C.D. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics (ACL): Stroudsburg, PA, USA, 2014; pp. 1532–1543.
30. Sarzynska-Wawer, J.; Wawer, A.; Pawlak, A.; Szymanowska, J.; Stefaniak, I.; Jarkiewicz, M.; Okruszek, L. Detecting formal thought disorder by deep contextualized word representations. *Psychiatry Res.* **2021**, *304*, 114135. [CrossRef] [PubMed]
31. Yin, K.; Ding, Z.; Dong, Z.; Ji, X.; Wang, Z.; Chen, D.; Li, Y.; Yin, G.; Wang, Z. Person re-identification method based on fine-grained feature fusion and self-attention mechanism. *Computing* **2024**, *106*, 1681–1705. [CrossRef]
32. Wang, Z.F.; Yuan, P.Y.; Cao, Z.Y.; Zhang, L.Y. Feature reduction of unbalanced data classification based on density clustering. *Computing* **2024**, *106*, 29–55.
33. Peters, C.C.; Van Voorhis, W.R. *Statistical Procedures and Their Mathematical Bases*; McGraw-Hill: New York, NY, USA, **1940**.
34. Shannon, C.E. A mathematical theory of communication. *Bell Syst. Tech. J.* **1948**, *27*, 379–423. [CrossRef]
35. Fisher, R.A. The use of multiple measurements in taxonomic problems. *Ann. Eugen.* **1936**, *7*, 179–188. [CrossRef]
36. Guyon, I.; Weston, J.; Barnhill, S.; Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach. Learn.* **2002**, *46*, 389–422. [CrossRef]
37. Ferri, F.; Pudil, P.; Hatef, M.; Kittler, J. Comparative study of techniques for large-scale feature selection. In *Pattern Recognition in Practice IV*; Machine Intelligence and Pattern Recognition; North-Holland: Amsterdam, The Netherlands, 1994; Volume 16, pp. 403–413. [CrossRef]
38. Tibshirani, R. Regression shrinkage and selection via the lasso. *J. R. Stat. Soc. Ser. B Stat. Methodol.* **1996**, *58*, 267–288. [CrossRef]
39. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
40. Mbona, I.; Eloff, J.H. Feature selection using Benford's law to support detection of malicious social media bots. *Inf. Sci.* **2022**, *582*, 369–381. [CrossRef]
41. Ilias, L.; Roussaki, I. Detecting malicious activity in Twitter using deep learning techniques. *Appl. Soft Comput.* **2021**, *107*, 107360. [CrossRef]
42. Cardaioli, M.; Conti, M.; Di Sorbo, A.; Fabrizio, E.; Laudanna, S.; Visaggio, C.A. It'sa matter of style: Detecting social bots through writing style consistency. In *Proceedings of the 2021 International Conference on Computer Communications and Networks (ICCCN)*; IEEE: New York, NY, USA, 2021; pp. 1–9.
43. Wu, Y.; Fang, Y.; Shang, S.; Jin, J.; Wei, L.; Wang, H. A novel framework for detecting social bots with deep neural networks and active learning. *Knowl.-Based Syst.* **2021**, *211*, 106525. [CrossRef]
44. 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*; AAAI Press: Washington, DC, USA, 2011; Volume 5, pp. 185–192.

45. Kim, S.H.; Park, S.Y.; Seo, H.; Woo, J. Feature selection integrating Shapley values and mutual information in reinforcement learning: An application in the prediction of post-operative outcomes in patients with end-stage renal disease. *Comput. Methods Programs Biomed.* **2024**, *257*, 108416. [CrossRef]
46. Gupta, S.; Singh, B. An intelligent multi-layer framework with SHAP integration for botnet detection and classification. *Comput. Secur.* **2024**, *140*, 103783. [CrossRef]
47. Hossain, M.A.; Saif, S.; Islam, M.S. A novel federated learning approach for IoT botnet intrusion detection using SHAP-based knowledge distillation. *Complex Intell. Syst.* **2025**, *11*, 422. [CrossRef]
48. Alhumaid, K.; Akour, I.; Salloum, S.A. Bot Detection and Credibility Score Analysis Using Random Forest and Network Analysis. In *Generative AI in Creative Industries*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 205–219.
49. 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]
50. 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 26th International Conference on World Wide Web Companion*; ACM: New York, NY, USA, 2017; pp. 963–972.
51. Feng, S.; Wan, H.; Wang, N.; Li, J.; Luo, M. Twibot-20: A comprehensive twitter bot detection benchmark. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*; ACM: New York, NY, USA, 2021; pp. 4485–4494.
52. Daouadi, K.E.; Rebaï, R.Z.; Amous, I. Bot detection on online social networks using deep forest. In *Proceedings of the Artificial Intelligence Methods in Intelligent Algorithms: Proceedings of 8th Computer Science On-line Conference 2019*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 28, pp. 307–315.
53. Beskow, D.M.; Carley, K.M. Bot conversations are different: Leveraging network metrics for bot detection in twitter. In *Proceedings of the 2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*; IEEE: New York, NY, USA, 2018; pp. 825–832.
54. Przybyła, P.; Soto, A.J. When classification accuracy is not enough: Explaining news credibility assessment. *Inf. Process. Manag.* **2021**, *58*, 102653. [CrossRef]
55. 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]
56. 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]
57. Diaz-Garcia, J.A.; Ruiz, M.D.; Martin-Bautista, M.J. A Comparative Study of Word Embeddings for the Construction of a Social Media Expert Filter. In *Proceedings of the International Conference on Flexible Query Answering Systems*; Springer: Berlin/Heidelberg, Germany, 2021; pp. 196–208.
58. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of Tricks for Efficient Text Classification. *arXiv* **2016**, arXiv:1607.01759.
59. Joulin, A.; Grave, E.; Bojanowski, P.; Douze, M.; Jégou, H.; Mikolov, T. FastText.zip: Compressing text classification models. *arXiv* **2016**, arXiv:1612.03651.
60. Strobl, C.; Boulesteix, A.L.; Zeileis, A.; Hothorn, T. Bias in random forest variable importance measures: Illustrations, sources and a solution. *BMC Bioinform.* **2007**, *8*, 25. [CrossRef]
61. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [CrossRef] [PubMed]
62. Strobl, C.; Boulesteix, A.L.; Kneib, T.; Augustin, T.; Zeileis, A. Conditional variable importance for random forests. *BMC Bioinform.* **2008**, *9*, 307. [CrossRef]
63. Venkatesh, B.; Anuradha, J. A review of feature selection and its methods. *Cybern. Inf. Technol.* **2019**, *19*, 3–26. [CrossRef]
64. 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*; ACM: New York, NY, USA, 2020; pp. 55–63.
65. Echeverría, 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*; ACM: New York, NY, USA, 2018; pp. 137–146.
66. Hays, C.; Schutzman, Z.; Raghavan, M.; Walk, E.; Zimmer, P. Simplistic collection and labeling practices limit the utility of benchmark datasets for twitter bot detection. In *Proceedings of the ACM web conference 2023*; ACM: New York, NY, USA, 2023; pp. 3660–3669.
67. Wu, J.; Ye, X.; Man, Y. Bottrinet: A unified and efficient embedding for social bots detection via metric learning. In *Proceedings of the 2023 11th International Symposium on Digital Forensics and Security (ISDFS)*; IEEE: New York, NY, USA, 2023; pp. 1–6.
68. Heidari, M.; Jones, J.H., Jr.; Uzuner, O. Online user profiling to detect social bots on twitter. *arXiv* **2022**, arXiv:2203.05966.
69. Fazil, M.; Sah, A.K.; Abulaish, M. Deepsbd: A deep neural network model with attention mechanism for socialbot detection. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 4211–4223. [CrossRef]

70. Rovito, L.; Bonin, L.; Manzoni, L.; De Lorenzo, A. An evolutionary computation approach for twitter bot detection. *Appl. Sci.* **2022**, *12*, 5915. [CrossRef]
71. Huang, H.; Tian, H.; Zheng, X.; Zhang, X.; Zeng, D.D.; Wang, F.Y. CGNN: A compatibility-aware graph neural network for social media bot detection. *IEEE Trans. Comput. Soc. Syst.* **2024**, *11*, 6528–6543. [CrossRef]
72. Chen, S.; Feng, S.; Songsong, L.; Zong, C.C.; Li, J.; Li, P. CACL: Community-Aware Heterogeneous Graph Contrastive Learning for Social Media Bot Detection. In *Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024*; Ku, L.W., Martins, A., Srikumar, V., Eds.; Association for Computational Linguistics: Bangkok, Thailand, 2024; pp. 10349–10360. [CrossRef]
73. Zhou, M.; Zhang, D.; Wang, Y.; Geng, Y.a.; Dong, Y.; Tang, J. Lgb: Language model and graph neural network-driven social bot detection. *arXiv* **2024**, arXiv:2406.08762.
74. Lei, Z.; Wan, H.; Zhang, W.; Feng, S.; Chen, Z.; Li, J.; Zheng, Q.; Luo, M. BIC: Twitter bot detection with text-graph interaction and semantic consistency. *arXiv* **2022**, arXiv:2208.08320.
75. He, B.; Yang, Y.; Wu, Q.; Liu, H.; Yang, R.; Peng, H.; Wang, X.; Liao, Y.; Zhou, P. Botdgt: Dynamicity-aware social bot detection with dynamic graph transformers. *arXiv* **2024**, arXiv:2404.15070.