From Online Behaviours to Images: A Novel Approach to Social Bot Detection*

Edoardo Di Paolo[1][0000−0001−9216−8430], Marinella Petrocchi[2][0000−0003−0591−877X], and Angelo Spognardi[1][0000−0001−6935−0701]

[1] Computer Science dept., Sapienza University of Rome, Italy
{dipaolo,spognardi}@di.uniroma1.it
[2] IIT-CNR, Pisa, Italy
marinella.petrocchi@iit.cnr.it

Abstract. Online Social Networks have revolutionized how we consume and share information, but they have also led to a proliferation of content not always reliable and accurate. One particular type of social accounts is known to promote unreputable content, hyperpartisan, and propagandistic information. They are automated accounts, commonly called bots. Focusing on Twitter accounts, we propose a novel approach to bot detection: we first propose a new algorithm that transforms the sequence of actions that an account performs into an image; then, we leverage the strength of Convolutional Neural Networks to proceed with image classification. We compare our performances with state-of-the-art results for bot detection on genuine accounts / bot accounts datasets well known in the literature. The results confirm the effectiveness of the proposal, because the detection capability is on par with the state of the art, if not better in some cases.

1 Introduction

With the advent of the internet and Online Social Networks (OSNs), production and fruition of information feature less mediated procedures, where content and quality do not always go through a rigorous editorial process [5, 15, 39]. Thus, although OSNs make our lives easier by giving us immediate access to information and allowing us to exchange opinions about anything, the danger of being exposed to false or misleading news is high [18, 27, 35]. The promotion of disinformation on OSNs has often been juxtaposed with the existence of automated accounts known as bots. As an example, Shao et al., in [33], have highlighted the role of Twitter bots, showing how bots were primarily responsible for the early

* This work was partially supported by project SERICS (PE00000014) under the NRRP MUR program funded by the EU - NGEU; by the Integrated Activity Project TOFFEe (TOols for Fighting FakEs) https://toffee.imtlucca.it/; by the IIT-CNR funded Project re-DESIRE (DissEmination of ScIentific REsults 2.0); by 'Prebunking: predicting and mitigating coordinated inauthentic behaviors in social media' project, funded by Sapienza University of Rome.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

spread of disinformation, interacting with influential accounts through mentions and replies.

The struggle between bots hunters and bots creators has been going on for many years now [6], and the actions of these automated accounts with malicious intent have influenced even the purchase of Twitter itself -just remember the $44 billion deal that went up in smoke precisely because of concerns about the unquantified presence of bots on the platform[3].

In this study, we provide a novel approach to bot detection, by leveraging the remarkable advancements in the field of image recognition [22, 25]. To the best of the authors' knowledge, no methodology or tool so far defined for bot detection leverage image recognition. In particular, we will exploit the potential of Convolutional Neural Networks (CNNs) to classify Twitter accounts as bots or not.

Based on the premise that automated accounts are often programmed to carry out spam and/or disinformation campaigns, numerous works in the literature have proposed detection approaches that leverage coordination and synchronism features of accounts, see, e.g. [4, 9, 47]. The intuition is that the online activities of a group of automated accounts -all devoted carrying out a certain strategy- are more similar to each other than those of genuine accounts. This is the leit motif from which the concept of *Digital DNA*, originally introduced by Cresci et al. in [11], and the detection technique known as *Social Fingerprinting* [8] came to life. The digital DNA of an online account represents the sequence of actions of that account and each action is associated to a symbol from a pre-defined alphabet. By associating each symbol with a color, it is possible to transform the sequence into an image, where each pixel represents the color of the corresponding symbol. The assumption that leads us to exploit image classification to perform bot detection is that images of bot accounts are similar to each other, and different from those of genuine accounts, given the different behavior of the two categories of accounts.

We thus propose an algorithm to transform sequences of digital DNA into images and we run pre-trained CNNs, such as VGG16, ResNet50, and WideResNet50 [20, 34] over the generated images. DNA sequences are extracted from Twitter accounts, both genuine and bot, of public datasets well-known in the bot detection literature. Where accounts' timelines are too short to produce good quality images, we have enhanced the latter, even turning features of the accounts other than Digital DNA into part of it.

Main Contributions: The main contributions of this work are as follows:

- Definition and implementation of a new approach to bot detection, based on image recognition;
- Validation of the approach by comparing our performance with state-of-the-art performances on publicly-released datasets: bot detection via image

[3] Elon Musk terminates $44B Twitter deal. Online: `https://nypost.com/2022/07/08/elon-musk-terminates-44-billion-twitter-deal/` August 8, 2022.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

recognition achieves the same performances as obtained in the literature, when not better.

We argue that the investigations here presented make the transition from sequence of actions to sequence of pixels for bots detection look promising. Of course, the literature is filled with more than good work in the field. Still, we find the approach itself interesting because it leverages well-established image recognition techniques. Thus, a way forward for further experimentation.

2 Related Work

Bot detection is a topic that began to be studied more than 10 years ago, when social networks became increasingly popular, and interests in spamming, spreading propaganda content, and increasing one's notoriety on the platforms grew tremendously. Different techniques have followed over the years, from classifying via traditional machine learning exploiting features in the accounts profile – the very first attempts in this direction are the papers by Mustafaraj and Metaxas [28] and Yardi et al. [45], both dated 2010 –, to using deep learning techniques. We therefore feel it is appropriate to list some of the work on the subject, without however intending to propose an exhaustive list.

Traditional machine learning approaches. Botometer is probably the most well-known tools in the literature for bot unveiling [43]; it is based on a supervised machine learning approach employing Random Forest classifiers. The last version, Botometer v4, has been recently shown to perform well for detecting both single-acting bots and coordinated campaigns [32]. v4 provides a useful lite version, BotometerLite[4], which does not interface with Twitter, but simply takes the tweet, retrieves the author, and does the necessary follow-up analysis. This light version only needs the information in the user profile to perform bot detection and hence can also process historical data published by accounts that are no longer active.

Over the years, there has been no limit in engineering accounts' features, to feed them to machine learning classifiers, e.g., the length of usernames, the reposting rate, some temporal patterns and the similarity of message contents based on Levenshtein distance [12], just to name a few.

Deep learning approaches. Hayawi *et al.* in [19] propose DeeProBot, where only some of the user profile features (the username's length and the number of followers) are exploited in order to classify single accounts with an LSTM (Long Short-Term Memory) network. Najari *et al.* in [29] use a GAN (Generative Adversarial Network) associated with a LSTM network. GANs generate bot samples to obtain more information about their behavior. RoSGAS (Reinforced and Self-supervised GNN Architecture Search) [44] is based on multi-agent deep reinforcement learning.

[4] `https://cnets.indiana.edu/blog/2020/09/01/botometer-v4/`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

In [23], the authors propose a deep neural network based on a LSTM architecture processing the texts of tweets, thus exploiting NLP techniques. Their intuition is that bot accounts produce similar contents; therefore, analyzing texts can help the classification. Authors of [40] propose a text-based approach using a bidirectional LSTM. Work in [41] presents a framework with deep neural networks and active learning to detect bots on Sina Weibo.

All of the cited works have been tested on publicly released bot datasets and achieve very good performances (greater than 0.9), considering standard classification metrics such as accuracy, precision, recall and Area Under the Curve.

Graph-based approaches. Detection techniques also take into account graph neural networks, where the social network is seen as a graph, where users are the nodes and the edge between two nodes represents a relationship between users, such as, e.g., a followship or retweet relationship. Thus, features derived from the social graph were considered along with profile and timeline fatures to train new models. An example is the work by Alhosseini *et al.* [1] which achieves very high performances, still on publicly released datasets [42].

Behavioral analysis. Approximately from 2014, a number of research teams, independently, proposed new approaches for detecting coordinated behavior of automated malicious accounts, see, e.g., [7, 47]. That line of research does not consider individual account properties, but rather properties in common with a group of accounts, like detection of loosely synchronized actions [4]. It is precisely in the context of the analysis of synchronism and coordination of the account behaviours that the idea of associating symbols with account actions arose, so that the timeline is represented as a string, so called Digital DNA [11]. The concept of Digital DNA is fundamental in the present work and will be introduced in the next section. Recently, Digital DNA has been re-analysed by Gilmary et al. in [16], where they measure the entropy of the DNA sequences. Even in this case, the detection performances result in very high values.

This brief roundup of work might lead one to think that bot detection is a solved task. Unfortunately, bots evolve faster than detection methods [8, 14], the latter are not suitable for detecting all kinds of bots [26], and existing datasets for doing training are inherently built according to peculiar accounts characteristics [30, 38].

We therefore conclude this section by pointing out how, perhaps, the task can never be solved in its entirety [6, 31], and that, since we still have room for investigation, relying on image detection and state-of-the-art tools in this regard seems to us to be a good track to take.

3 Useful Notions

3.1 Digital DNA

The biological DNA contains the genetic information of a living being and is represented by a sequence which uses four characters representing the four nu-

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

cleotide bases: A (*adenine*), C (*cytosine*), G (*guanine*) and T (*thymine*). Digital DNA is the counterpart of biological DNA and it encodes the behaviour of an online account. In particular, it is a sequence consisting of L characters from a predefined alphabet $\mathbb{B}$:

$$\mathbb{B} = \{\ \sigma_1, \sigma_2, \sigma_3, ..., \sigma_N\} \tag{1}$$

In Equation 1 each σ is a symbol of the alphabet and a digital DNA sequence will be defined as follow:

$$s = (\sigma_1, \sigma_2, ..., \sigma_n),\ \sigma_i \in \mathbb{B}\ \forall\ i = 1, ..., n. \tag{2}$$

Each symbol in the sequence denotes a type of action. In the case of Twitter, a basic alphabet is formed by the 3 actions representing the types of tweets:

$$\mathbb{B} = \left\{ \begin{array}{c} A = \text{tweet}, \\ C = \text{reply}, \\ T = \text{retweet} \end{array} \right\} \tag{3}$$

According to the type of tweets, it is thus possible to encode the account timeline, which could be, e.g., the following $s = ACCCTAAACCCCCCCTT$. Strings of digital DNA were compared to each other in [9]: the longer the *longest common substring* of a group of accounts, the more likely that group is made up of accounts programmed to complete a similar task, i.e., the more likely those accounts are automated accounts.

3.2 Convolutional Neural Networks

Given the recent and noteworthy [17,22,25,36] advancements in the field of Convolutional Neural Networks (CNNs), we asked ourselves whether these networks could be used to classify Twitter accounts into bot / human.

CNNs are typically composed of three layers: convolutional layers, pooling layers, and fully connected layers. The convolutional layer is the fundamental component of a CNN and it requires most of the computation. The input is an image and the dimension of the input image changes depending on whether it is grayscale or colored. Combined with the convolutional layer, there is the "filter" which is a matrix of small size. From the convolution operation, we have in output a "filtered" image which is a sequence of dot products between the input pixels and the filter. Afterward, an activation function can be applied to the output. It is also possible to optimize the convolution layer through some hyperparameters, such as the "stride" and the "padding". The former represents the amount of movement of the filter over the input, the latter is the process of padding the border of the input. The second type of layer is the "pooling" layer. A pooling layer is used to downsample the given input. There are two main types of pooling: max-pooling and average-pooling. The third type of layer is the "Fully-Connected" (FC) layer, also known as a "dense" layer. The neurons in a FC-layer receive input from all the neurons in the previous layer.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

4 From Digital DNA to Images

To the best of our knowledge, no approaches in the literature take advantage of image classification to classify social bots. The aim is transforming each account's DNA sequence into an image. Given the similarity in the sequences of bots' actions with respect to those of genuine accounts, the intuition is that an image classifier might work well in the bot detection task.

The literature offers some DNA-to-image conversion algorithms [21, 24]. We tried experimenting with these conversion algorithms, but we did not get significant results since these algorithms are for real DNA strings. Thus, we decided to propose an ad-hoc conversion algorithm, which transforms a digital DNA sequence into a bidimensional object.

Algorithm 1 shows the pseudocode for passing from Digital DNA to an image. Since CNNs expect images of the same size, we first consider the string of maximum length and check whether the length is a perfect square. If not, we consider the perfect square closest to and strictly largest than the maximum length. By doing so, it is possible to transform all the strings to images of equal size[5].

After arbitrarily deciding a RGB color to assign to each symbol in the alphabet, the image is colored pixel by pixel based on the coors assigned to the correspondent symbol. The coloring is done as long as the length of the input string is not exceeded; therefore, if the sequence is not the one with the maximum length, this will result in a black part of the image. All images created are in grayscale; we tried also with colored images, but there was no significant improvement in the final results.

5 Datasets

This section introduces the datasets on which we tested our detection technique. The datasets are all publicly available.

5.1 Cresci-2017

Firstly introduced in [8], this dataset consist of bots and genuine accounts. The kind of bots are various, like bots engaged in online political discussions, bots promoting specific hashtags, and bots advertising Amazon products. In our study, we evaluated 991 bots and 1,083 genuine accounts for a total of 2,074 samples.

The first step of the procedure is the generation of the DNA sequences for each account in the dataset. We rely on the alphabet in Section 3.1(Equation 3), which considers three symbols, associated to three basic activities on the Twitter platform: Tweet, Retweet, Reply. After the generation of the DNA strings, we apply the algorithm in Algorithm 1 to generate the images.

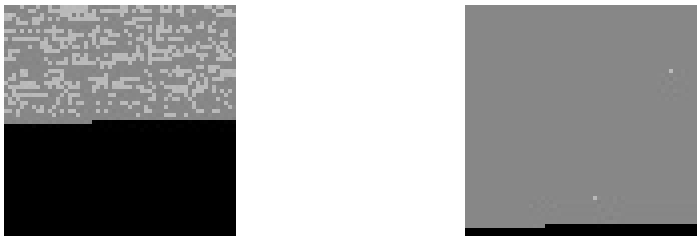
[5] As an example, strings as long as 10000 characters are represented by images of size 100x100.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

Algorithm 1 From Digital DNA to image: Pseudocode

Input: List of DNA sequences
Output: DNA images

$n \leftarrow$ Length of the longest DNA sequence
if n is a perfect square **then**
    $L \leftarrow \sqrt{n}$
else
    $L \leftarrow$ `get_closest_square_number(n)`
end if
$P \leftarrow$ dict with symbols and colors
for `each DNA sequence` **do**
    $I \leftarrow$ `create_image(width=L, height=L)`
    for `row in range(L)` **do**
        for `col in range(L)` **do**
            $k \leftarrow$ `(row * L) + col`
            if $k < n$ **then**
                $I[row, col] \leftarrow P[DNA[k]]$
            end if
        end for
    end for
end for



(a) (b)

Fig. 1: Representation as images of a genuine (left) and bot (right) account belonging to Cresci-2017.

Figure 1 show two images, representing a genuine and a bot account, *resp.* belonging to the Cresci-2017 dataset. Some noise in Figure 1a distinguishes this account from that of the bot (Figure 1b). Intuitively, a CNN is able to pick up these differences and, thus, classify the accounts in the correct way.

5.2 Cresci-Stock 2018

First introduced by Cresci et al. in [10], this dataset consists of both genuine and automated accounts tweeting so-called 'cashtags', i.e., specific Twitter hashtags that refer to listed companies. Part of the automated accounts has been found to act in a coordinated fashion, in particular by mass retweeting cashtags of low

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

(a) (b)

Fig. 2: Human and bot accounts on Cresci-Stock-2018 dataset.

capitalization companies. In our study, we used 6,842 bots and 5,882 genuine users, for a total of 12.724 labeled samples.

In Figure 2, it is possible to see the noise which distinguishes a genuine account (Figure 2a) from a bot account (Figure 2b).

In this case, the image representing the bot is almost completely white, due to the homogeneity of the actions it performs on the social network and due to the choice of the colors used for the different pixels in creating the images.

5.3 TwiBot20

Firstly introduced in [13], TwitBot20 is a very large dataset with almost 230k Twitter accounts. Of these, the authors provide a total of 11,746 labeled samples, 6,561 bots and 5,185 genuine accounts. The dataset covers diversified bots and genuine users 'to better represent the real-world Twittersphere' and, at the time of the publication of the article presenting it, it represented 'the largest Twitter bot detection benchmark'.

In TwiBot20, bot sequences of activities are very similar to those of genuine accounts. This of course makes the images generated by the sequences similar to each other and there is limited information to highlight specific behavioral patterns. Furthermore, TwiBot20 features accounts with a maximum of 200 tweets per user; therefore, the images are quite small (15x15). We attempted to enlarge the images, but the results were not good. Details are in Section 6.

6 Experiments and Results

For the experiments, we use PyTorch Lightning [6] to produce readable and reproducible code and it allows to spend less time on engineering the code. We also adopt WanDB [3] to keep track of metrics. Regarding the loss function, we consider the cross entropy (Equation 4).

$$Loss = -(y \log(p) + (1 - y) \log(1 - p)) \tag{4}$$

[6] `https://github.com/PyTorchLightning/pytorch-lightning`

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

The upperbound to the number of epochs is set to 50. However, we use `EarlyStopping` to monitor the accuracy (or loss) on the validation set: if it the accuracy does to increase (*resp.*, the loss does not decrease) for a predetermined number of epochs, the training stops. Each dataset tested is randomly splitted into training, testing and validation. The only exception is TwiBot20, where the authors of the dataset give this split. We evaluate the classification performances in terms of well-known, standard metrics, such precision, recall, F1 (the harmonic mean of precision and recall), and Matthew Correlation Coefficient (MCC) (i.e, the estimator of the correlation between the predicted class and the real class of the samples). The results achieved in this study are noticeable since they, in some cases, improve the state of the art. In general, we tried several pre-trained models, but the best results were achieved by networks based on the ResNet50 model. During the training phase, we carefully monitored the loss so as to be sure that there were no overfitting problems, and, thus, the model learned to classify correctly.

Comparison between state-of-art results and those by the image classification proposal for Cresci-2017 In the paper introducing TwitBot-20 [13], the authors consider two other datasets, Cresci-2017, already introduced by us above, and PAN-19[7]. To all 3 datasets, the authors of TwitBot-20 apply state-of-the-art detection methods, to evaluate the difference in classification performances. The best result obtained on the Cresci-2017 dataset by [13] is reported in Table 1, first row. The same table, in the second row, shows the performance results obtained by applying our method based on image classification, with ResNet50, where the loss decreases to 0.114. From that table, we can note how our perfomance results are equal to state-of-the-art results, with a slight improvement in MCC.

Table 1: Performances' comparisons on Cresci-2017: state-of-art *vs* image classification.

	Cresci 2017			
Metric	Accuracy	Recall	F1	MCC
Feng et al.	0.98	-	0.98	0.96
Image classification	0.98	0.98	0.98	**0.98**

Figure 3 shows the training and validation losses: the two losses have similar trends, and they decrease until they stabilize, after a number of epochs. Since the model behaves similarly in both the validation and training set, there is no overfitting [46].

Comparison between state-of-art results and those by the image classification proposal for Cresci-Stock In this case, the results in [2] will be taken as a refer-

[7] https://pan.webis.de/clef19/pan19-web/author-profiling.html

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

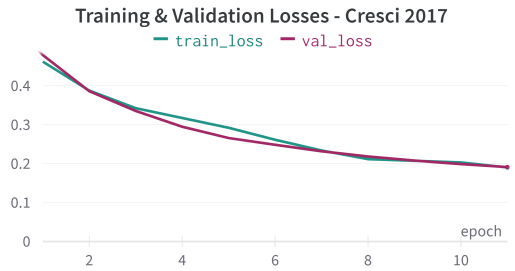


Fig. 3: Training and validation losses for Cresci-2017.

Table 2: Performances' comparisons on Cresci-Stock: state-of-art *vs* image classification. Results of *Antenore et al.* are taken from the Table 4 of [2].

	Cresci stock 2018			
Metric	Accuracy	Recall	F1 score	MCC
Antenore et al.	0.77	0.96	0.82	-
Image classification	**0.89**	0.88	**0.89**	**0.78**

ence. The comparison between the results are reported in Table 2, and, as it is possible to see, the images approach improved the *accuracy*, the *F1 score* and the *MCC*.

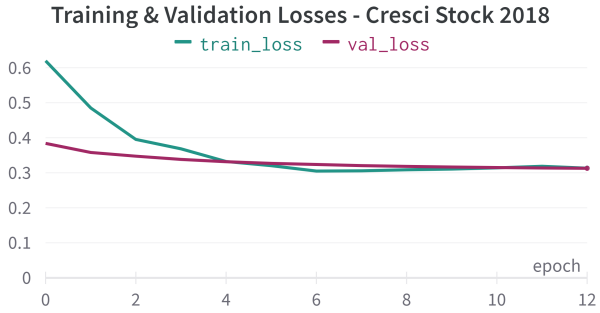
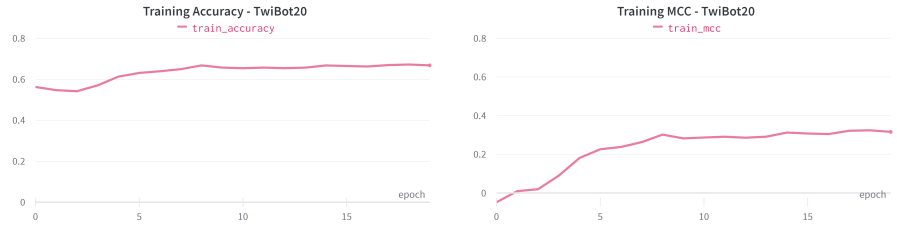


Fig. 4: Training and validation loss for Cresci-Stock.

In Figure 4, the two losses have a similar macroscopic behavior; in the training phase the loss stabilizes in fewer epochs, which is due to the larger number of samples used (60% in training, 30% in validation). This trend rules out overfitting [46].

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42



(a) Training accuracy. (b) Training MCC.

Fig. 5: Training accuracy and MCC considering the images originated from the TwitBot20 account timelines

Table 3: Performances' comparison on TwitBot20: state-of-art *vs* image classification *vs* image classification where the image is enriched with SuperTML features [37].

	TwiBot20			
Metric	Accuracy	Recall	F1 score	MCC
Feng et al.	0.81	-	0.85	0.67
Images approach	0.67	0.66	0.61	0.34
Images approach with SuperTML	0.81	0.80	0.80	0.67

Image classification for the TwiBot20 accounts After applying the Algorithm 1 to the timelines of the TwiBot20 accounts and after the CNN training phase, the classification performances are disappointing due to the limited set of tweets per user. It can be seen in Figure 5b and Figure 5a in which the accuracy does not improve so much compared to the initial phase, and MCC stabilizes between 0.3 and 0.4. The exact numerical results are in Table 3, second row.

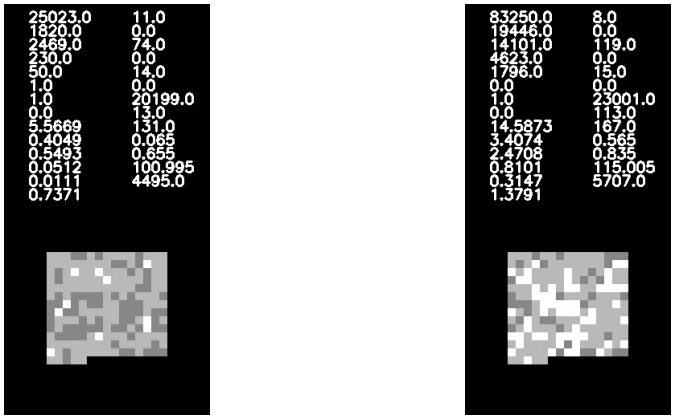
Thus, we decide to use more account features and attach them to the user timeline. Interestingly enough, the article in [37], by Sun et al., proposes an algorithm to represent tabular data as images and then proceeds with image classification. Classification achieves state-of-art results on both large and small datasets, like the Iris dataset[8] and the Higgs Boson Machine Learning[8].

Given the effective approach of [37], we enlarge our feature set: the digital DNA plus all the features listed in Table 4. Table 3 shows the resulting images, for a genuine and a bot account. The third row in Table 3 shows the performance results of the classification, when the image is formed with the enlarged feature set.

Finally, Figure 7 shows the trends of the training and validation losses, which stabilize around 0.6. Even in this case, the losses have very similar behavior, thus, the model is not overfitting.

[8] Iris dataset homepage, Higgs Boson Machine Learning challenge on Kaggle.

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42



(a) (b)

Fig. 6: (a): image format for a genuine account; (b): image format for a bot account. Both images have been created following the SuperTML algorithm, proposed in [37]. The upper part of the image is the list of features also reported in Table 4. The lower part of the image is the representation of the Digital DNA.

Table 4: List of features used.

Features
statuses_count, followers_count, friends_count, listed_count, default_profile, favourites_count, profile_use_background_image, verified, followers_growth_rate, friends_growth_rate, favourites_growth_rate, listed_growth_rate, followers_friends_rate, screen_name_length, screen_name_digits_count, description_length, description_digits_count, name_length, name_digits_count, total_tweets_chars_count, total_urls_in_tweets, total_mentions_in_tweets, urls_tweets_rate, mentions_tweets_rate, chars_tweets_rate, account_age

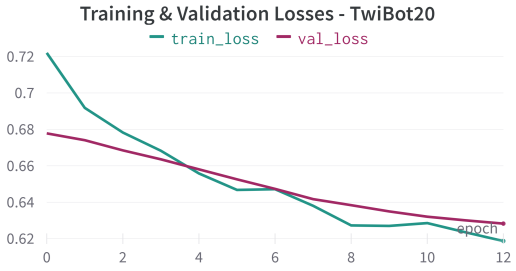


Fig. 7: Training and validation losses for TwitBot20 Image Classification + SuperTML

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

7 Conclusions

Research in bot classification is still open, mainly due to the continuous evolution of these kind of accounts. This work proposed a novel method for the task, based on image classification. The proposed approach has been proven aligned with state-of-art results. Since it is tough to acquire fully representative benchmark datasets (e.g., social platforms often block scraping, the APIs have a limited number of calls), the natural way to follow is to achieve full advantage of the available data. In the case of TwitBot20, for example, the original dataset offers numerous other pieces of information that were not exploited in the present work to obtain the images, such as, e.g., the network of interactions between accounts. As future work, we might consider exploiting this extra information to better evaluate the proposed approach.

References

1. Ali Alhosseini, S., Bin Tareaf, R., Najafi, P., Meinel, C.: Detect me if you can: Spam bot detection using inductive representation learning. In: 2019 World Wide Web Conference, Companion. p. 148–153. WWW '19, ACM (2019). https://doi.org/10.1145/3308560.3316504
2. Antenore, M., Rodriguez, J.M.C., Panizzi, E.: A comparative study of bot detection techniques with an application in Twitter Covid-19 discourse. Social Science Computer Review (2022). https://doi.org/10.1177/08944393211073733
3. Biewald, L.: Experiment tracking with weights and biases (2020), https://www.wandb.com/
4. Cao, Q., Yang, X., Yu, J., Palow, C.: Uncovering large groups of active malicious accounts in online social networks. In: ACM SIGSAC Conference on Computer and Communications Security. pp. 477–488. ACM (2014)
5. Ceron, A.: Internet, news, and political trust: The difference between social media and online media outlets. Journal of computer-mediated communication **20**(5), 487–503 (2015)
6. Cresci, S.: A decade of social bot detection. Commun. ACM **63**(10), 72–83 (2020)
7. Cresci, S., Di Pietro, R., Petrocchi, M., Spognardi, A., Tesconi, M.: DNA-inspired online behavioral modeling and its application to spambot detection. IEEE Intelligent Systems **31**(5), 58–64 (2016)
8. 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: 26th International Conference on World Wide Web Companion. pp. 963–972. ACM (2017). https://doi.org/10.1145/3041021.3055135
9. 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. **15**(4), 561–576 (2018)
10. Cresci, S., Lillo, F., Regoli, D., Tardelli, S., Tesconi, M.: $FAKE: Evidence of spam and bot activity in stock microblogs on Twitter. In: ICWSM (2018). https://doi.org/10.1609/icwsm.v12i1.15073
11. Cresci, S., Pietro, R.D., Petrocchi, M., Spognardi, A., Tesconi, M.: Dna-inspired online behavioral modeling and its application to spambot detection. IEEE Intell. Syst. **31**(5), 58–64 (2016). https://doi.org/10.1109/MIS.2016.29, https://doi.org/10.1109/MIS.2016.29

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

12. Efthimion, P.G., Payne, S., Proferes, N.: Supervised machine learning bot detection techniques to identify social twitter bots. SMU Data Science Review **1**(2), 5 (2018)
13. Feng, S., Wan, H., Wang, N., Li, J., Luo, M.: Twibot-20: A comprehensive Twitter bot detection benchmark. In: CIKM '21. pp. 4485–4494. ACM (2021). https://doi.org/10.1145/3459637.3482019
14. Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The rise of social bots. Commun. ACM **59**(7), 96–104 (Jun 2016)
15. Gangware, C., Nemr, W.: Weapons of Mass Distraction: Foreign State-Sponsored Disinformation in the Digital Age. Park Advisors (2019)
16. Gilmary, R., et al.: Dna-influenced automated behavior detection on Twitter through relative entropy. Scientific Reports **12**, 8022 (2022)
17. Gu, J., et al.: Recent advances in convolutional neural networks. Pattern Recognition **77**, 354–377 (2018). https://doi.org/https://doi.org/10.1016/j.patcog.2017.10.013
18. Guo, B., Ding, Y., Yao, L., Liang, Y., Yu, Z.: The future of misinformation detection: New perspectives and trends. CoRR **abs/1909.03654** (2019), http://arxiv.org/abs/1909.03654
19. Hayawi, K., et al.: DeeProBot: a hybrid deep neural network model for social bot detection based on user profile data. Soc. Netw. Anal. Min. **12**(1), 43 (2022). https://doi.org/10.1007/s13278-022-00869-w
20. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. CoRR **abs/1512.03385** (2015), http://arxiv.org/abs/1512.03385
21. Jeffrey, H.: Chaos game representation of gene structure. Nucleic Acids Research **18**(8), 2163–2170 (04 1990). https://doi.org/10.1093/nar/18.8.2163
22. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
23. Kudugunta, S., Ferrara, E.: Deep neural networks for bot detection. Information Sciences **467**, 312–322 (2018). https://doi.org/10.1016/j.ins.2018.08.019
24. LA, S., et al.: DNA sequence recognition using image representation. Research in Computing Science **148**, 105–114 (2019). https://doi.org/10.13053/rcs-148-3-9
25. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A ConvNet for the 2020s. In: Computer Vision and Pattern Recognition. pp. 11966–11976 (2022). https://doi.org/10.1109/CVPR52688.2022.01167
26. Mazza, M., Avvenuti, M., Cresci, S., Tesconi, M.: Investigating the difference between trolls, social bots, and humans on Twitter. Computer Communications **196**, 23–36 (2022)
27. Meel, P., Vishwakarma, D.K.: Fake news, rumor, information pollution in social media and web: A contemporary survey of state-of-the-arts, challenges and opportunities. Expert Systems with Applications **153** (2020). https://doi.org/https://doi.org/10.1016/j.eswa.2019.112986
28. Mustafaraj, E., Metaxas, P.T.: From obscurity to prominence in minutes: Political speech and real-time search. In: Web Science: Extending the Frontiers of Society On-Line (2010)
29. Najari S., Salehi M., F.R.: Ganbot: a gan-based framework for social bot detection. Soc. Netw. Anal. Min. **12**(4) (2022). https://doi.org/10.1007/s13278-021-00800-9, https://doi.org/10.1007/s13278-021-00800-9
30. Olteanu, A., Castillo, C., Diaz, F., Kıcıman, E.: Social data: Biases, methodological pitfalls, and ethical boundaries. Frontiers in Big Data **2**, 13 (2019)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42

31. Rauchfleisch, A., Kaiser, J.: The false positive problem of automatic bot detection in social science research. PLoS One **15**(10) (2020)
32. Sayyadiharikandeh, M., et al.: Detection of novel social bots by ensembles of specialized classifiers. In: CIKM '20: The 29th ACM International Conference on Information and Knowledge Management. pp. 2725–2732. ACM (2020)
33. Shao, C., et al.: Anatomy of an online misinformation network. Plos one **13**(4), e0196087 (2018)
34. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) Learning Representations (2015)
35. Suarez-Lledo, V., Alvarez-Galvez, J.: Prevalence of health misinformation on social media: Systematic review. J Med Internet Res **23**(1), e17187 (Jan 2021). https://doi.org/10.2196/17187, http://www.jmir.org/2021/1/e17187/
36. Sultana, F., Sufian, A., Dutta, P.: Advancements in image classification using convolutional neural network. In: 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). pp. 122–129. IEEE (2018), http://arxiv.org/abs/1905.03288
37. Sun, B., Yang, L., Zhang, W., Lin, M., Dong, P., Young, C., Dong, J.: Supertml: Two-dimensional word embedding and transfer learning using imagenet pretrained CNN models for the classifications on tabular data. CoRR **abs/1903.06246** (2019), http://arxiv.org/abs/1903.06246
38. Tan, Z., Feng, S., Sclar, M., Wan, H., Luo, M., Choi, Y., Tsvetkov, Y.: BotPercent: Estimating Twitter bot populations from groups to crowds. arXiv:2302.00381 (2023)
39. Valkenburg, P.M., Peter, J.: Comm research—views from europe| five challenges for the future of media-effects research. International Journal of Communication **7**, 19 (2013)
40. Wei, F., Nguyen, U.T.: Twitter bot detection using bidirectional long short-term memory neural networks and word embeddings. In: Trust, Privacy and Security in Intelligent Systems and Applications. pp. 101–109 (2019). https://doi.org/10.1109/TPS-ISA48467.2019.00021
41. 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. Knowledge-Based Systems **211** (2021). https://doi.org/https://doi.org/10.1016/j.knosys.2020.106525
42. Yang, C., Harkreader, R., Gu, G.: Empirical evaluation and new design for fighting evolving twitter spammers. IEEE Transactions on Information Forensics and Security **8**(8), 1280–1293 (2013). https://doi.org/10.1109/TIFS.2013.2267732
43. Yang, K., Varol, O., Davis, C.A., Ferrara, E., Flammini, A., Menczer, F.: Arming the public with AI to counter social bots. CoRR **abs/1901.00912** (2019), http://arxiv.org/abs/1901.00912
44. Yang, Y., Yang, R., Li, Y., Cui, K., Yang, Z., Wang, Y., Xu, J., Xie, H.: RoSGAS: Adaptive social bot detection with reinforced self-supervised GNN architecture search. Trans. on the Web (2022). https://doi.org/10.1145/3572403
45. Yardi, S., Romero, D., Schoenebeck, G., et al.: Detecting spam in a Twitter network. First Monday (2010). https://doi.org/10.5210/fm.v15i1.2793
46. Ying, X.: An overview of overfitting and its solutions. In: Journal of physics: Conference series. vol. 1168, p. 022022. IOP Publishing (2019)
47. Yu, R., He, X., Liu, Y.: GLAD: Group anomaly detection in social media analysis. ACM Transactions on Knowledge Discovery from Data (TKDD) **10**(2), 1–22 (2015)

ICCS Camera Ready Version 2023
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-35995-8_42