Impersonation on social media: a deep neural approach to identify ingenuine content

Koosha Zarei, Reza Farahbakhsh, Noel Crespi, Gareth Tyson

Impersonation on Social Media: A Deep Neural Approach to Identify Ingenuine Content

Koosha Zarei*, Reza Farahbakhsh*, Noël Crespi*, Gareth Tyson†
*Institut Polytechnique de Paris, Télécom SudParis Evry, France.
{koosha.zarei, reza.farahbakhsh, noel.crespi}@telecom-sudparis.eu
†Queen Mary University of London, United Kingdom. gareth.tyson@qmul.ac.uk

***Abstract*—Impersonators are playing an important role in the production and propagation of the content on Online Social Networks, notably on Instagram. These entities are nefarious fake accounts that intend to disguise a legitimate account by making similar profiles and then striking social media by fake content, which makes it considerably harder to understand which posts are genuinely produced. In this study, we focus on three important communities with legitimate verified accounts. Among them, we identify a collection of 2.2K impersonator profiles with nearly 10k generated posts, 68K comments, and 90K likes. Then, based on profile characteristics and user behaviours, we cluster them into two collections of 'bot' and 'fan'. In order to separate the impersonator-generated post from genuine content, we propose a Deep Neural Network architecture that measures 'profiles' and 'posts' features to predict the content type: 'bot-generated', 'fan-generated', or 'genuine' content. Our study shed light into this interesting phenomena and provides interesting observation on bot-generated content that can help us to understand the role of impersonators in the production of fake content on Instagram.**

***Index Terms*—Impersonators; Fake Profile; Fake Content; Fake Engagement; Bot; Instagram; Social Media.**

I. Introduction

Impersonation is where (sometimes malicious) users create social media accounts mimicking a legitimate account [1]. Fr example, impersonators or imposters maybe accounts that pretend to be someone popular or a representative of a known brand, company, *etc.* Such impersonators are found on all major social media platforms. Instagram is widely used by celebrities, influencers, businesses, and public figures with different levels of popularity. Although many impersonators may be innocuous, there also exists malicious fake accounts. These often have clear plans, where they make accounts appear more popular than they are, produce pre-planned untrustworthy content, perform brand abuse or generate fake engagement [2]. Therefore several lawsuits have taken place in the United State (along with other countries), where criminal impersonation is a crime. It involves assuming a false identity with the intent to defraud another or pretending to be a representative of another person or organisation [3]. However, identifying such activities is often slow and laborious — hence, developing techniques for automated detection would have real value to social media companies. In this paper, we aim to identify impersonator-generated content in Instagram. Towards that end, we pick three different and important communities with verified genuine accounts inside each. Through the pool of collected public content, by using the methodology presented in [4], we identify a set of 2.2K impersonator accounts. Next, by using unsupervised learning techniques, we find two notable clusters: (*i*) '*Cluster 0 - Bots*' that represent bot entities, and (*ii*) '*Cluster 2 - Fans*' which represent fan entities. In this study, bots are fake accounts or social bots that tend to mimic the real user and accomplish a specific purpose [5] and interacts with humans on social media [6]. In contrast, fans are (semi-) human-operated accounts that are created and maintained by a fan or devotee about a celebrity, thing or particular phenomenon. We then use these clusters to create necessary labels for building and training a Deep Neural Network to predict post types: (*i*) bot-generated, (*ii*) fan-generated, or (*iii*) genuine content. The contribution of this study can be summarised as follow:

- We assemble a novel dataset containing the content and activities of impersonators in three leading communities: Politicians, Sport Starts and Musicians.
- We present a practical approach to cluster impersonators and generate content labels based on profile characteristics and user behaviours.
- We propose a Deep Neural Network architecture in order to detect and predict impersonator-generated posts and genuine content.

II. Methodology & Dataset

A. Definition and Taxonomy

Bots: are (semi-) automatic agents that are designed to accomplish a specific purpose [5] and automatically produce content and interacts with humans on social media [6]. Bots are normally defined with the condition of mimicking human behaviour [7].

Impersonator or Imposter: is someone on social media who builds a profile using the information of another legitimate account and pretends to be that entity or copies the behaviour/actions of that profile [2].

Profile Similarity: We use this term to indicate whether there is any similarity or correlation between two Instagram profiles. Similarity can be in (*i*) text features [8] such as username, full name, or biography *e.g.* '*@barackobama*' and '*@barack__obama*', or (*ii*) profile photos (if the same person exists in both photos). The '*Similarity Level*' could be high (similar in all metrics), low (just in one metric), or

between. An example of the genuine Theresa May account and her impersonator with a high degree of similarity is shown in Figure 1. In [1] [4] we introduced the problem of impersonation and discussed the identification methods. Then, we uncovered unknown groups of impersonators and examined their behaviours. For example, fan pages have a higher number of followers and are completely public pages. But bots, have very fewer followers and publish a lot of posts in a shorter period of time. Then, in [2] we studied the comments they generated under the post of genuine figures in details. For example, bots produce much higher duplicated comments than others and give likes (passive reaction) faster. Eventually, in this study, we divide impersonators into two broad types of public accounts:

(1) **Bot Impersonator (Bot)**: these public fake accounts or social bots tend to mimic the real user and generally generate specific content. First, from profile characteristics, bots are usually simple accounts that use default Instagram settings: no full name, no biography, and sometimes no profile photos. The follower count is low and they follow a lot of other accounts. From similarity viewpoint (compared to a genuine user), bots have weak profile similarity degrees: they have no similar profile photo and have low similarity in username, full name, or biography. From activity viewpoint, bots receive very limited engagement (like or comment) per post, are lazy in publishing stories, are so active in giving comments and likes to others, and the rate of issuing duplicated comments is high. Existing bots vary in sophistication. Some bots are very simple and merely re-publish posts, whereas others are sophisticated and can even interact with human users or post comment. In this study, '*Bot Impersonator*' and '*bot*' terms are interchangeable.

(2) **Fan Impersonator (Fan)**: is a (semi-) human-operated account that is created and maintained by a fan or devotee about a celebrity, thing or particular phenomenon. From profile perspective, fans have a greater follower number than bots, are completely public accounts, have a biography, and usually use a URL. From impersonation viewpoint, fans have higher profile similarity in photo, username, full name, and biography metrics. From behaviour viewpoint, fans are interested in publishing posts and stories, are more productive than bots, receive higher engagement within their posts (both like and comment), and the owner barely shares self-generated content. From managing viewpoint (who controls the page), we can divide fans into two different types (Figure **??**): (*i*) A fan page which is regulated by 'human'. In this situation, there is no automation movement and all content and activities are published by a human. (*ii*) A fan page which is regulated by 'human and bot'. In this type, page owner which is a human usually use some automation and bot services to gain attention. For example, using a bot to comment or like on related pages.

B. Case Study Accounts

To seed our analysis, we select a set of 15 ground-truth verified accounts from three communities: *politicians*, *sports stars*, and *musicians (celebrities)*. We pick these communities

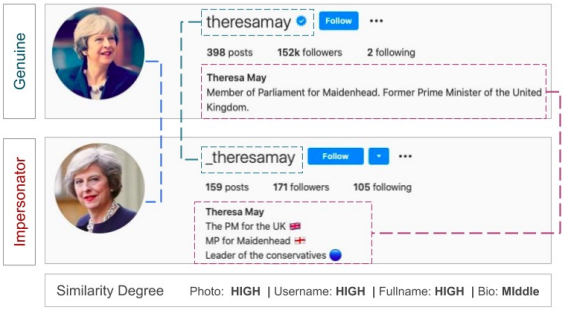

Fig. 1: Identifying Impersonators through profile similarity.

to compare the impersonation problem in divided societies. For each community, we select the top 5 most popular verified accounts manually, then we confirm the popularity by [9]:

Politicians: Donald J. Trump (@*realdonaldtrump*), Barack Obama (@*barackobama*), Emmanuel Macron (@*emmanuelmacron*), Boris Johnson (@*borisjohnsonuk*), and Theresa May (@*theresamay*). ***Sports Stars:*** Leo Messi (@*leomessi*), Cristiano Ronaldo (@*cristiano*), Rafael Nadal (@*rafaelnadal*), Roger Federer (@*rogerfederer*), and Novak Djokovic (@*djokernole*). ***Musicians:*** Lady Gaga (@*ladygaga*), Beyonce (@*beyonce*), Taylor Swift (@*taylorswift*), Adele (@*adele*), and Madonna (@*madonna*).

TABLE I: Use Cases and Corresponding Hashtags

Politician		Sports Stars		Musician	
D. Trump	*#donaldtrump*	L. Messi	*#leomessi*	L. Gaga	*#ladygaga*
B. Obama	*#barackobama*	C. Ronaldo	*#cristianoronaldo*	Beyonce	*#beyonce*
E. Macron	*#emmanuelmacron*	R. Federer	*#rogerfederer*	T. Swift	*#taylorswift*
B. Johnson	*#borisjohnson*	R. Nadal	*#rafaelnadal*	Madonna	*#madonna*
T. May	*#theresamay*	N. Djokovic	*#novakdjokovic*	Adele	*#adele*

C. Data Collection

Genuine Accounts: First, we collect posts of our 15 genuine case studies (listed in section "II-B") which are published between October 2018 and January 2020. Posts contain publicly available information including caption, hashtags, image/video, number of likes, number of comments, location, time, and tagged list. 1.3K posts across the three communities has been collected during the campaign. We use the crawler presented in [1].

Identifying Impersonators: To obtain a set of impersonators, we configure a crawler to collect public posts that contain associated hashtag with the name of each account (Table I) between September 2019 and January 2020. For example, in Trump, we gather posts include the *#donaldtrump* tag. Next, based on the methodology that we presented in [1], we measure the profile similarity of the publishers to identify impersonators across case studies. The methodology is based on the Instagram profile similarity and we consider major profile metrics such as *username (text), full name (text), biography (text), profile photo (image), follower count, followee count, media count, and account age*. (*i*) For text metrics, we use the Cosine Similarity technique [8] and we define the minimum threshold to 30%. (*ii*) To measure the photo similarity, we use a convolutional neural network face detection in [10]. We compare the face of all accounts (if exist) to the face of the genuine users (*e.g.* R. Federer) and if the same person is detected, we mark it as similar photos. Eventually, if an

account has at least 30% similarity in one of the text metrics or has a similar profile photo, we consider it as an impersonator. Otherwise, it is a non-similar account (*not* impersonator) and we exclude it from the dataset. In total, we discover 1.6K impersonators with different levels of similarity.

Followers/Followees: We next crawl the follower and followee list of each impersonator from the previous phase (October 2018 to January 2020). As it is infeasible to collect *all* followers/followees, we define a limitation of 1K for followers and 500 for followees. At the same time, we examined the profile similarity of them to see if they are impersonator or not. Finally, we have 2.3K impersonators.

Posts: We crawled the 50 most recent posts published by the impersonator. Furthermore, we gather impersonators' (*i*) profile information, (*ii*) number of comments received on posts, and (*iii*) number of likes attracted on posts. This task was running simultaneously between October 2018 and January 2020.

Validation: We finally manually inspect the profiles of the impersonators to confirm they are impersonators. We filter any incorrectly identified impersonators alongside their posts. 36 Impersonators were identified incorrectly (1.5% of the total population), and 42 accounts (1.8%) change the application of the page or sell their account at some point during the measurement period. In total, we obtain nearly 68K comments and 90K likes from 10K posts of 2.2K impersonators (Table II).

TABLE II: Summary of Dataset

Community	Imposter	post	comment	like
Politician	36%	30%	36%	35%
Sport player	34%	30%	34%	40%
Musician	30%	40%	30%	25%
Total	**2.2K**	**10K**	**68K**	**90K**

Ethics: In line with Instagram policies and ethical consideration on user privacy defined by the community, we only collect publicly available data through public API excluding any potentially sensitive data.

D. Data Pre-Processing

Pre-Processing Some features require pre-processing: (*i*) For caption and Profile Biography, we remove all punctuation marks, stopwords and convert them to lowercase characters. We then filter words that contain fewer than three characters, and words are stemmed to reduce to their root forms. (*ii*) We then remove and covert all emojis and emoticons to word format. Then we replace URLs with 'website', emails with 'email', new lines with 'line', and phone numbers with 'phones'. (*iii*) We break down each Hashtag and Username into its constituent words, *e.g.* "*makeamericagreatagain*" contains 4 meaningful words: "make", "america", "great", and "again" [11]. (*iv*) From posts and profile biographies, we extract hashtags (#) and mentions (@) into separated lists. (*v*) Wherever possible, we extract the text from post image thumbnail using Tesseract OCR [12] and apply text pre-processing steps. The spaCy [13] is used for French Language Modeling.

TABLE III: Real Accounts vs. Impersonators

use case	follower		followee		avg. #comment per post		avg. #like per post	
	Imp (avg)	*real account*	*Imp (avg)*	*real account*	*Imp*	*real account*	*Imp*	*real account*
D. Trump	528	16M	1.1K	8	27.14	19.5K	690.14	340K
B. Obama	256	2.5M	446	14	40.00	13.5K	1.4K	1M
E. Macron	435	1.5M	738	91	12.45	3.8K	302.03	65K
B. Johnson	431	367K	318	254	11.78	600	274.14	15K
T. May	312	157K	253	1	2.21	350	54.25	5.6K
Ch. Ronaldo	432	197M	832	445	12.16	35K	1.6K	5.5M
L. Messi	447	140M	650	227	13.08	28K	2.8K	4.1M
R. Nadal	121	8.4M	513	65	12.17	2.5K	768.23	290K
R. Federer	189	7.1M	479	71	9.45	2.9K	670.12	400K
N. Djokovic	148	6.6M	236	777	6.67	1.5K	320.05	220K
Lady Gaga	7.2K	39M	653	46	5.46	19.5K	219.46	1.1M
Beyonce	130	138M	701	0	3.92	25.8K	353.18	2.9M
Taylor Swift	2.4K	125M	1.3K	0	4.84	0.0*	177.83	1.8M
Adele	5.3K	33M	459	0	3.76	12.7K	291.15	1.3M
Madonna	6.6K	14.7M	842	243	4.74	1.8K	134.45	98K

*T. Swift disabled comments.

III. Who are Impersonators?

We start by making some primary analysis. Table III presents some of the fundamental differences between real accounts and impersonators. Impersonators tend to have few followers, but they follow many others. Normally, they do this to develop a network of relevant accounts (other impersonators) and increase their followers. Also, impersonators have a lower engagement rate.

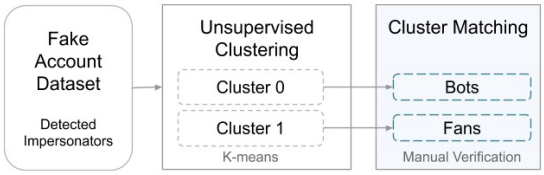

Fig. 2: The process of discovering impersonators.

Clustering. To find the potential hidden impersonators, we do clustering. The whole process is explained in Figure 2. First, we use the impersonator dataset from section II as input and based on profile characteristics and behaviour activities, we perform unsupervised learning. We experiment with a number of clustering methods, including K-means, Gaussian Mixture Modeling, and Spectral Clustering, finding similar results. Our feature list consist several features listed in Table IV. This identifies two clusters (the optimal number is obtained from the Elbow Method). The two derived clusters are highly diverse in profile characteristics and publishing behaviour (Table V). For the rest of this study, results are based on the K-means algorithm. Based on manual confirmation we match discovered clusters with types of impersonators defined in section II-A. Inspection of these clusters reveals two clear populations:

TABLE IV: Clustering Feature Set.

similarity username	avg received like	follower
similarity full name	avg hashtag length	followee
similarity biography	avg caption length	media count
similarity photo	avg received comment	private
external url	account age	verified
MSF*	LSF*	

*The most and least number of features that have similarity.

Cluster 0 - Bot: We believe this cluster captures bot entities (Section "II-A") that exist to achieve specific tasks. In this study, bots are fake entities that are programmed to publish pre-defined content as posts, use a particular network of hashtags, and target specific issues. Bots have a quite low

similarity in all profile metrics (less than 20%) and the number of followers is almost 6 times fewer than fans (Table V). However, the rate of post-distribution is higher in bots. One of the important metrics is the received attention per post (passive or active) and bots earned nearly half of fans (almost 10 comments and 770 likes).

Cluster 1 - Fan: Based on assessing characteristics, we acknowledge that this cluster represents Fans. Fans spread content regarding a genuine figure (in favour of or against). There is nearly 50% similarity in the username, 40% in the full name, 20% in biography, and 70% similarity in profile photos. Moreover, they hold similarity at most in 3 metrics. The number of followers is higher than the bots (avg. 101.6K vs. 16.5K) and on average, each post got 24 comments and nearly 1.6K likes (Table V).

TABLE V: Characteristics of the clusters.

Metrics	Fans	Bots
avg. username similarity per imp*	**0.49**	0.13
avg. full_name similarity per imp	**0.40**	0.18
avg. bio similarity per imp	0.25	0.18
avg. photo similarity per imp	**0.71**	0.17
the Least number of features that have similarity	1	1
the Most number of features that have similarity	3.32	1.53
avg. follower per imp	**101.6K**	16.5K
avg. followee per imp	757	927
avg. media count per imp	**808**	679
avg. received comment per post	**24.15**	10.01
avg. received like per post	**1.6K**	774

*Impersonator

Manual inspection for validation. To validate the correctness of the proposed clustering, from each cluster we pick 80% of profiles and check each one manually. Based on the definitions (Section II-A), 112 accounts were identified incorrectly. As we were not sure if those accounts represent a bot character or a fan entity, we recognized them as outliers and excluded from the clusters. The rest of this study is based on these validated impersonators.

IV. Identifying Impersonator Content

We next exploit the above dataset to explore the possibility of automatically identifying impersonator posts. We believe that a bot, as a fake identity, also produces untrustworthy content and fake engagements. Likewise, fan pages, in some cases may distribute fake content *e.g.* a political fan page may publish rumours. So, we use the labelled data from the previous section and present a DNN classifier to distinguish content types. This classifier can predict whether a post is impersonator-generated (fan or bot) or genuine-generated. Note that we do not consider the question of classifying the veracity of information shared by the accounts.

A. Data Preparation

Dataset Overview. For classification, we use the post dataset obtained after clustering which is described in Section II. This dataset consists of 10K post from 2.2K impersonators across 3 communities. Since we conduct manual annotation of impersonators, we are confident that posts are labelled correctly (pre-processing steps are discussed in Section II-C).

Over-Sampling. Our dataset is highly unbalanced: 31% genuine, 45% fan-generated, and 34% bot-generated post content. To solve this problem, we use the combination of Synthetic Minority Over-sampling Technique (SMOTE) [14] and Random Under-sampling algorithm [15]. So, we produce similar examples from the minor class to increase the total number and, meanwhile, we under-sample the major class and randomly remove some samples. The final dataset contains an equal amount of samples from class types. This helps us to increase the final accuracy by 8.5%.

TABLE VI: Feature Set used in Deep Neural Network.

Post Features		Publisher Features	
Feature	**Type**	**Feature**	**Type**
caption text	*text*	similarity username	*numeric*
caption topics (LDA)	*text*	similarity fullname	*numeric*
post hashtag	*text*	similarity bio	*numeric*
tagged users in post	*text*	profile biography	*text*
like count	*numeric*	similarity photo	*numeric*
comment count	*numeric*	follower/followee/post	*numeric*
tagged users count	*numeric*	full name	*text*
mention users count	numeric	biography	*text*
hashtag count	numeric	username	*text*
overall sentiment of caption	numeric	following followers ratio [16]	*numeric*
overall sentiment of hashtag	numeric	followers posts ratio	*numeric*
media type (image or video)	*numeric*	bio emoji count	*numeric*
emoji count	*numeric*	bio hashtag count	*numeric*
url/website exist	*numeric*		*numeric*
date	*numeric*		

Feature Engineering. We build a set of features from post metadata and profile metrics that help us to train the proper model (Table VI). We break the feature list into two principal categories: "*post features*" which comprises all features that are obtained from the content of the post such as number of likes, the caption, *etc.* And "*publisher features*" that are extracted from the profile of the publisher profile. To prepare the feature set, we directly use some features such as numbers. However, some others are derived from the content. For example, the account age is taken from the date of the first post and the profile similarities are calculated previously in section II-C. Then, to do text vectorization, the caption text, user biography, and other text metrics are vectorized using Keras Tokenizer [17] class with 30000 num_words. This class allows vectorizing a text corpus, by turning each text into either a sequence of integers.

Proposed DNN Architecture. Then, we propose a Deep Neural Network architecture that exploits CNN, LSTM, BERT and Dence Layers to process post content and profile metadata (Figure 3). The workflow is as follows:

(1) First, in the input layer, we extract and pre-process all features that are listed in Table VI. This architecture accepts two inputs types: (*i*) text content (*e.g.* post caption, hashtags, profile bio) which we combine them into a single corpus. (*ii*) the metadata features (*e.g.* like, comment, follower, followee) that come from both profile and post content and then are transformed into a single vector.

(2) Next, to transform the text into a form amenable for processing, we adopt a pre-trained language model, Bidirectional Encoder Representations from Transformers (BERT) [18]. This, results in an output vector by BERT (vectorized text) and then given as input to a CNN layer.

(3) Then, the tokenized output of the BERT layer passes through a Convolution Neural Networks. This network contains 1D CNN with ReLU activation function (and 128 filters and a kernel size of 6) followed by a Dropout Layer (value of 0.2) for regularization, then a 1D Pooling Layer.

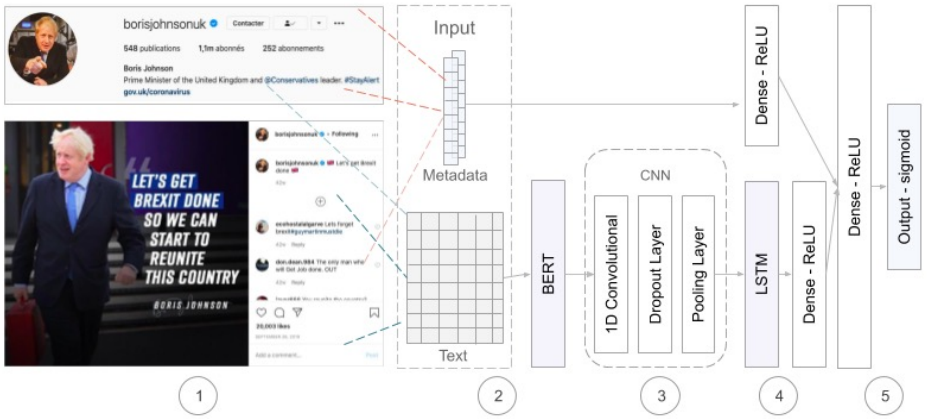

Fig. 3: The proposed Deep Neural Network architecture to detect impersonator content.

(4) Then, (*i*) the result of CNN layer connects to a LSTM layer which processes vectorized text data and outputs a single 32-dimensions vector that is then fed forward through a ReLU activated Dense layer of size 16. (*ii*) Meanwhile, numerical metadata passes through a Dense Layer with ReLU activation of size 16.

(5) Finally, we concatenate the output of the text and metadata layers into a single vector (size 32) that is then fed forward through a Dense layer with ReLU activation function and then an Output Layer which forms the type of the post (bot, fan, genuine). We develop this model using Tensorflow and Keras Functional API [17].

We pick a random split of 75% (training set) and 25% (test set) and run with 10-Fold Cross-Validation. The Accuracy, Precision, Recall, and F1-Score results are listed in Table VII. We compare the proposed classifier with a tradition Random Forest Classifier. The traditional RF Classifiers give approximately 77% in all metrics (text tokenized using TF-IDF). First, we do classification using the proposed DNN architecture with only 'post content' (CNN + LSTM), and we observe an increase in overall result by nearly 2% (Accuracy 78%). Then we re-run the classifier with both '*post content*' and '*profile metadata*' (CNN + LSTM). This helps to improve by almost 4.5% (Accuracy 83%). Finally, we add the BERT layer to our architecture (BERT + CNN + LSTM). This step additionally assists us to improve the overall efficiency by almost 4%, and we achieve the accuracy of 86% in detecting post type.

TABLE VII: Performance of the proposed architecture

Model	Accuracy	Precision	Recall	F1
Random Forest Classifier	0.76	0.78	0.77	0.76
Proposed DNN (post)	0.78	0.79	0.76	0.78
Proposed DNN (post + profile)	0.83	0.82	0.83	0.82
Proposed DNN (post + profile) + BERT	**0.86**	**0.85**	**0.86**	**0.85**

V. Conclusion

This study focuses on impersonators problem and the challenge of identifying the impersonator-generated content on Instagram. First, by the help of clustering we recognised two clusters and based on their characteristics, we clustered them as Fans and Bots. Then, in order to detect what do they publish, we introduced a DNN which can correctly classify posts as 'bot-generated', 'fan-generated', or 'genuine' content. The results of this study help community on better understanding the phenomena of bot-generated content in social media.

References

[1] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. Deep dive on politician impersonating accounts in social media. In *2019 IEEE Symposium on Computers and Communications (ISCC) (IEEE ISCC 2019)*, Barcelona, Spain, June 2019.

[2] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. How impersonators exploit instagram to generate fake engagement?, 2020.

[3] uslegal. https://definitions.uslegal.com/c/criminal-impersonation/, 2019.

[4] Koosha Zarei, Reza Farahbakhsh, and Noel Crespi. Typification of impersonated accounts on instagram. In *2019 IEEE 38th International Performance Computing and Communications Conference (IPCCC) (IPCCC 2019)*, London, United Kingdom (Great Britain), October 2019.

[5] Christian Grimme, Mike Preuss, Lena Adam, and Heike Trautmann. Social bots: Human-like by means of human control?, 2017.

[6] Emilio Ferrara, Onur Varol, Clayton Davis, Filippo Menczer, and Alessandro Flammini. The rise of social bots. *Commun. ACM*, 59(7), 2016.

[7] Stefan Stieglitz, Florian Brachten, Björn Ross, and Anna-Katharina Jung. Do social bots dream of electric sheep? a categorisation of social media bot accounts, 2017.

[8] Baoli Li and Liping Han. Distance weighted cosine similarity measure for text classification. In *Intelligent Data Engineering and Automated Learning – IDEAL 2013*, pages 611–618, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[9] Top 1000 Instagram Influencers Ranking. https://hypeauditor.com/top-instagram-sports/?source=imh&source2=imh-ig, 2019.

[10] Face Recognition. Face recognition. github.com/ageitgey/face_recognition, January 2020.

[11] Word Ninja Github. https://github.com/keredson/wordninja., 2019.

[12] R. Smith. An overview of the tesseract ocr engine. In *Proceedings of the Ninth International Conference on Document Analysis and Recognition - Volume 02*, ICDAR '07, page 629–633, USA, 2007. IEEE Computer Society.

[13] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[14] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, Jun 2002.

[15] Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017.

[16] Kai-Cheng Yang, Onur Varol, Pik-Mai Hui, and Filippo Menczer. Scalable and generalizable social bot detection through data selection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1096–1103, Apr 2020.

[17] François Chollet et al. Keras. https://keras.io, 2015.

[18] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.