# Impostor GAN: Toward Modeling Social Media User Impersonation with Generative Adversarial Networks

Masnoon Nafees, Shimei Pan, Zhiyuan Chen, and James R. Foulds(✉)

University of Maryland Baltimore County, Baltimore, MD, USA
{masnoon1,shimei,zhchen,jfoulds}@umbc.edu

**Abstract.** The problem of "fake accounts" on social media has gained a lot of attention recently because of the role they play in generating/propagating misinformation, manipulating opinions and interfering with elections. Among them, "impersonators" or "impostors" are those fake social media accounts that mimic the posts and behavior of a targeted person (e.g., a politician or celebrity) or brand. In this preliminary investigation, we study a GAN-based framework in which an impostor aims to produce realistic social media posts which pass for being created by the target person while the detector aims to identify the posts by the impostors/impersonators. The impostor and the detector are co-trained until an equilibrium is reached, where the impostor is good at mimicking the posts of the target person and the detector is good at identifying the posts of an impostor. Our model allows us to achieve both impersonation and its detection, and also to study how this adversarial scenario might unfold. We applied our method to a Twitter dataset to study its performance. The results demonstrate that our proposed model is promising in generating and detecting impostors' posts.

**Keywords:** Social media · Impersonation · Adversarial networks

## 1 Introduction

Fake social media accounts are pervasive. According to Facebook's 2020 Community Standards Enforcement Report, the company estimates that around 5% of its accounts were fake in Q4 of 2020.[1] *Impersonator* or *impostor* accounts, which aim to assume the identity of another social media user, are a particularly problematic type of fake account. Social media impersonation is a widespread problem, motivated by fraud (e.g. "brandjacking"), misinformation, propaganda, brand abuse, and the manipulation of popularity and engagement metrics [10].

In this paper, we propose to study social media impersonation from a machine learning perspective. We propose a method called Impostor GAN, which simulates an impersonator, an impersonation detector, and the adversarial game in

[1] https://transparency.facebook.com/community-standards-enforcement.


S. Sarkadi et al. (Eds.): DeceptECAI 2020/DeceptAI 2021, CCIS 1296, pp. 157–165, 2021.
https://doi.org/10.1007/978-3-030-91779-1_11

which these two entities compete with each other, using a Generative Adversarial Network (GAN). This allows us to achieve impersonation and its detection, and to study how this adversarial scenario might play out in an idealized setting, which could provide insights into the real-world impersonation problem. Our experimental results, although preliminary, were quite encouraging.

## 2 Background and Related Work

There is a large body of work on social bot detection [7,8]. The first-generation social bots were quite simple, with few social connections and limited ability to automatically generate posts. These bots are relatively easy to detect. For example, [7] employs supervised machine learning to accurately detect these bots. Recently, to avoid detection, social bots have become increasingly sophisticated. They are carefully engineered to be like humans. These bots are very difficult to detect for both algorithms and humans [9]. Recently, [10] focused on detecting impostors, who try to mimic a specific person. Our work differs in that we study the impostor, the impostor detector, and their interaction, using a GAN.

Generative Adversarial Networks (GANs) [1] have become quite popular recently. As deception detection is intrinsically adversarial, we apply GANs to generate and detect posts from impostors. The idea of the GAN is to play a min-max game between a generator network and a discriminator network. The discriminator tries to distinguish between real and fake data. The generator tries to beat the discriminator. The game continues until the generator can eventually generate realistic samples that fool the discriminator. More specifically, we base our approach on the WGAN [2]. The WGAN tries to address several issues such as vanishing gradients in the original GAN. WGANs clip the weights in the discriminator to enforce the Lipschitz constraint. They use different techniques like linear activation functions, and most importantly, the Wasserstein loss function which is based on the earth mover distance. WGAN-GP [3] tries to further improve the WGAN. The authors proposed a method that penalizes the gradient norm with respect to its input of the discriminator or critic instead of weight clipping. This method generally works better than WGANs and GANs. Due to these advantages, in this paper we have used a Wasserstein GAN-based formulation to develop our Impostor GAN model.

## 3 Methodology

Our proposed Impostor GAN model consists of three networks: two generators and one discriminator. The two generators aim to produce ambiguous tweets for impersonation, and the discriminator aims to detect impersonation. The first generator generates data points in an embedding space that are semantically similar to a specific user's tweets (*user 1*, a.k.a. *user*, e.g. the target user to be impersonated), but which are difficult to distinguish from a second user's tweets (*user 2*, a.k.a. *otheruser*, e.g. an impersonating user). The second generator does the reverse: it produces data points that are semantically similar to the second

user's tweets, but which are difficult to distinguish from the first user's tweets. The generators thereby try to impersonate users by selecting points in the overlap region common to both users, i.e. ambiguous tweets which could have come from either user.

To circumvent the challenging task of generating realistic content from either user, we simplify the problem to the task of selecting from a set of existing social media posts from an impersonating user. The impersonating user could be chosen to be a specific other user who we want to make look like the target user, or to the content from multiple (or all) other users. In our future work, we plan to generate realistic posts via transformer models such as GPT-3 [11].

The objective of the discriminator is to detect this deception by distinguishing between the user 1 and user 2 tweets created by the two generators. It maximizes the score of the target user's tweets while minimizing the score of the other users' tweets. The discriminator and two generators are iteratively updated in turn. Eventually, the Impostor GAN aims to achieve a Nash equilibrium where the two generators will generate points in the overlapping region of the target user's tweets and the other users' tweets. Such selected tweets will be difficult to distinguish, and hence can be used for impersonation.

More formally, our proposed Impostor GAN model consists of two generators, $G_1$ and $G_2$, and a discriminator $D$. As shown in the architecture diagram in Fig. 1, and algorithm pseudocode in Algorithm 1, $G_1$ and $G_2$ take inputs from a common latent space $z$. Let $D_1$ be the data set for a specific user and $D_2$ be the data set for other user/users. $D_1$ and $D_2$ both generate data points in the embedding space, but $G_1$ generates points similar to the specific user's data but $G_2$ generates points similar to the other user(s)' data. Since these data points may not exactly match an existing data point (e.g., a tweet), we use nearest neighbor search to find a data point from $D_1$ that is closest to the data point generated by $G_1$. Similarly the point from $D_2$ that is closest to the data point generated by $G_2$ is also returned. We use the Ball-Tree algorithm to speed up nearest neighbor search and cosine as the distance metric [4].

Let $N_1$ be the set of closest data points from $D_1$ and $N_2$ be the closest points from $D_2$. $N_1$ and $N_2$ are passed to the discriminator $D$. We use a WGAN-style formulation [2]. We formulate the model as a min-max game:

$$min_{G_{user,otheruser}} max_D \left[ \mathbb{E}[D(G_1(x_i^{N_{user}}))] - \mathbb{E}[D(G_2(x_i^{N_{otheruser}}))] \right] . \quad (1)$$

The objective of the discriminator is to maximize the score of the data point in $D_1$ (i.e., the specific user's data) and minimize the score of a data point from $D_2$ (i.e., other users' data):

$$max_D \left[ \mathbb{E}[D(G_1(x_i^{N_{user}}))] - \mathbb{E}[D(G_2(x_i^{N_{otheruser}}))] \right] . \quad (2)$$

Following the WGAN, we used a linear activation function in the last layer of the discriminator. While in typical image applications of WGANs it is common to bound the generator's output between [−1,1] using tanh in the final layer, in our case we also used a linear final layer for the generators. $G_1$'s objective is to

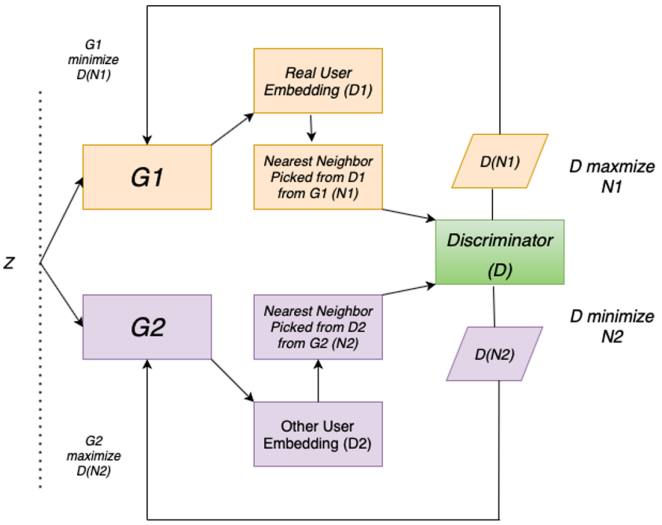



**Fig. 1.** Impostor GAN model architecture

**Algorithm 1:** Training of Impostor GAN

**Input**: Initial Generator $G1$, Generator $G2$, Discriminator $D$, Real User Embedding Data $D1$, Other User Embedding Data $D2$, batch size = x, $d$ as number of updates for the discriminator
**Parameter**: Optimizer = ***Adam***, $D$'s learning rate $\alpha_d = 0.00002$,
$G_1$ and $G_2$'s learning rate $\alpha_g = 0.00001$,
batch size = 32, distance metric = *cosine*
**Result**: $N_1$ and $N_2$ contain most overlapped data
Input of $N_1$, $N_2$ in D;
**while** *Until Converge* **do**
  **for** $i = 0;\ i < d;\ i++$ **do**
    Update $D$ to maximize Equation 2 via Adam
  **end**
  Update $G_1$ to minimize Equation 3 via Adam
  Update $G_2$ to maximize Equation 4 via Adam
**end**

minimize the score given by the discriminator for data points from the *user*, and $G_2$'s objective is to maximize the discriminator score for the *other user*'s data:

$$min_{G_1}[\mathbb{E}[D(N(x_i^{user}))]] \ , \tag{3}$$

$$max_{G_2}[\mathbb{E}[D(N(x_i^{otheruser}))]] \ . \tag{4}$$

The two generators and the discriminator are updated based on the gradients of the above objective functions, e.g. via the Adam optimizer.

In practice, following [2] we update the discriminator more times than the generators. We found that training the discriminator 3 times more than the generators gave us better results while using a batch size of 32 for both generators. As we have multiple neural networks we also focused on how we can achieve a local Nash equilibrium. We noticed that applying different learning rates for the discriminator and generators provides more meaningful results. We applied the *Two Time-Scale Update Rule* [5] by using a lower learning rate for the generators than the discriminator with the Adam optimizer. Slower learning rates on the generators help the networks to adjust based on the discriminator's feedback.

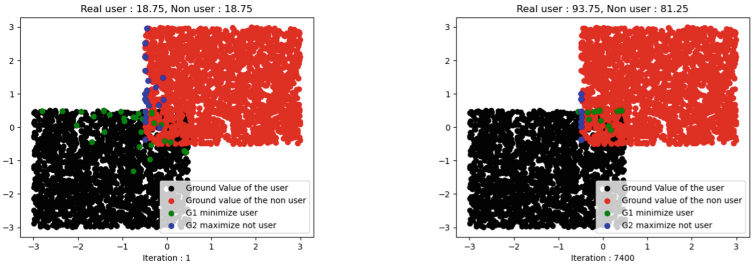
## 4 Dataset Description

For the experiment we selected a publicly available tweets data set from Donald Trump and Hillary Clinton [6]. We chose this dataset because these individuals' relatively well-known personalities make the results easier to interpret, and because there is a lot of public interest in any insights into their behavior and the 2016 election which might arise from the study. We made Trump tweets as the target *user* to be impersonated and Clinton tweets as the *other user* whose tweets will be used for impersonation, although note that the model is symmetric and performs impersonations for both users. Both data sets contains around 3000 tweets each. We followed standard text cleaning procedures to clean the text, remove any stop words, numeric values, etc. Then, the cleaned text data was converted into 50-dimensional embeddings. We used a Glove model to get word embeddings and doc2vec to convert the tweets into embeddings. We also generated a synthetic dataset to help understand and visualize the model's behavior and its ability to find the overlap region. The dataset contains 1000 2-dimensional *user* data points uniformly distributed in the range of $-3$ to 0.5 and 1000 instances for *other user* uniformly distributed in the range of $-0.5$ to 3.0.

## 5 Experiments

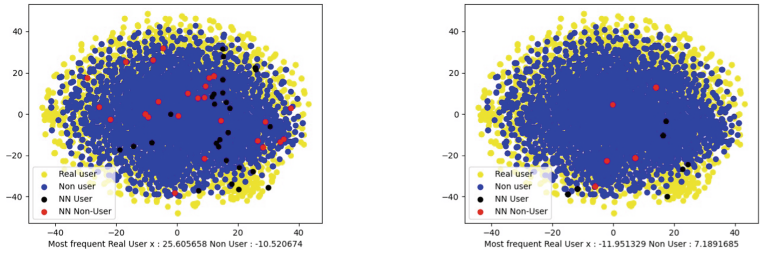
**Results for Synthetic Data:** Figure 2 *(a)* and Fig. 2 *(b)* show data points in $D_1$, $D_2$, $N_1$, and $N_2$ for the synthetic data set at iteration 1 and 7400, respectively. From Fig. 2 *(a)*, we can see that the two rectangle uniform distribution values are overlapped in $-0.5$ to 0.5 region. Green and Blue values are the nearest neighbor values picked by the two generators. Only 18.75% for both user and non user points picked by the generators lies in overlap region as the other points in $N_1$ and $N_2$ are scattered in the whole plot. In iteration 7400, we see that most of the data points selected by Impostor GAN are in the overlap region. This behavior is indicative of success, as points in the overlap region cannot easily be distinguished by the discriminator. This means that the three networks have converged in such a way that the selected data points will be ambiguous.

**Results for Twitter Data:** We then move to the real Twitter dataset. Unlike for image data, where we can see the quality of the generated images, in our




(a) Iteration 1 (b) Iteration 7400

**Fig. 2.** Results on synthetic data




(a) Iteration 1 (b) Iteration 6000

**Fig. 3.** Results on Twitter dataset ($t$-SNE projection)

case it is difficult to visually observe whether the method is working correctly. To gain some understanding of this numeric data experiment, we used $t$-SNE to convert the 50-dimensional embedding data into 2D for visualization purposes. The plot contains the embedding data for both $D_1$ and $D_2$ and the nearest neighbor data points. From Fig. 3 *(a)*, we can see that in iteration 1, the chosen nearest neighbor values are scattered. The most frequent values for both user and non user are also far apart from each other. After the model ran for 6000 iterations, Fig. 3 *(b)* shows that the most frequently selected user and non users points are closer than in the first iteration. Below are some sample tweets from Trump and Clinton. First, as a baseline approach, instead of using the Impostor GAN we report tweets selected based on the shortest distance between them and an embedding from the other user:

**Shortest Distance Baseline, *Trump*:**

- *"@markgruber1960: @megynkelly @realDonaldTrump That's why he is so successful. He is driven to succeed" True!*
- *Congratulation to Adam Scott and all of the folks at Trump National Doral on producing a really great WGC Tournament. Amazing finish!*

**Shortest Distance Baseline, *Clinton*:**

- *We have to build an economy that works for everyone, not just those at the top. #DebateNight https://t.co/XPTvh4Dovf*
- *Donald Trump says he "cherishes women". Just not if they're working and pregnant. https://t.co/sd9KHSvQlO https://t.co/MQeMLNtuNG*

We now report tweets selected by Impostor GAN after 6000 iterations.

**Impostor GAN, *Trump*:**

- *In Hillary Clinton's America - things get worse. #TrumpPence16 https://t.co/WdHbnhhCbW*
- *Hillary could lose to Trump in Democratic New York #MakeAmericaGreatAgain #Trump2016 https://t.co/fQR48CVIbt"*

**Impostor GAN, *Clinton*:**

- *No wonder Donald Trump is hiding his tax returns. #debatenight https://t.co/gcvsadMwHJ*
- *"Candidate Trump opposes equal pay, paid leave, and Planned Parenthood. President Trump? Same. https://t.co/H85Ud5jyOe"*

From the model's chosen tweets, we can see that our model has picked Trump's tweets that refer to Clinton, and Clinton's tweets that refer to Trump. It makes sense that these tweets are chosen since the other user's name is likely an effective obfuscation of the tweet author's identity in embedding space. On the other hand, given their understanding of the context of the 2016 election, a human adversary could likely still distinguish the author of these tweets based on the sentiment expressed toward the other person. Note however that this is a relatively challenging scenario. Based on the tweets in this data set that the authors have inspected, the authorship of the majority of the tweets is generally easy inferred, making it quite difficult to choose tweets which comprise effective impersonation against a human adversary in this context. The behavior of the method, in which the chosen tweets refer to the other user, does appear to show promise in a more typical scenario in which the human adversary does not have substantial contextual domain knowledge about the users, or where the adversary uses a machine learning model to distinguish the tweets' authorship.

## 6 Evaluation

The Impostor GAN's generator, which aims to achieve impersonation, can be said to succeed if it selects social media posts in the overlapping region of its two input users' posts' embeddings. To evaluate this, we have adapted the idea of calculating Frechet Inception Distance (FID) [5], which is commonly used to evaluate GANs for generating images. The FID calculates the distance between embeddings for real and generated images based on the overlap of their distributions, which are modeled as multivariate normal. The images are embedded using

the encoding layer of Inception, a large model for image classification. We instead evaluate using the Frechet distance for doc2vec embeddings of the tweets chosen by the Impostor GAN. If the model works as intended, the Frechet distance between the tweets selected by the two generators should decrease. We compared our approach to a baseline which simply calculates the shortest distance between the tweets. From Fig. 4, we can see that as the number of iterations increases, the Frechet distance typically goes down. Finally, the Frechet distance for the trained Impostor GAN was substantially better than the baseline (Table 1).

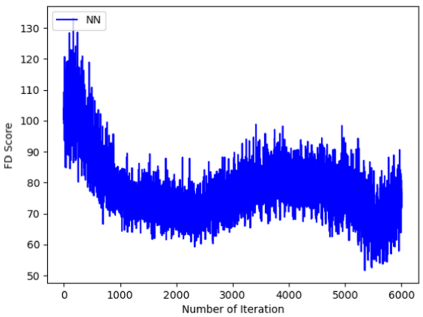



**Fig. 4.** Frechet Distance vs training iterations for Impostor GAN

**Table 1.** Frechet Distance (FD) for Shortest Distance Baseline and Impostor GAN (Lower is Better)

| Method | FD |
|---|---|
| Shortest Distance Baseline | 168.3 |
| Impostor GAN | 70.5 |

## 7 Conclusion

We have proposed Impostor GAN, a method for performing impersonation and also impersonation detection based on social media data such as tweets, and potentially, Facebook, Instagram, email, etc. The method used a novel GAN-based formulation involving a combination of three neural networks and a nearest neighbor procedure. Our preliminary experiments with our proposed model showed promising results. Both quantitative and qualitative results on synthetic 2D data and real Twitter data, including a Frechet distance evaluation, show that our model can converge into the overlap region of the data distributions for two users. In future work we plan to extend the model to generate new impersonating social media posts using transformer models such as GPT-3, instead of simply selecting from existing tweets. We will further study the implications of our model regarding the real-world adversarial competition between social media impostors and those who would detect them.

## References

1. Goodfellow, I.J., et al.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint arXiv:1701.07875 (2017)

3. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of Wasserstein GANs. arXiv preprint arXiv:1704.00028 (2017)
4. Muflikhah, L., Baharudin, B.: Document clustering using concept space and cosine similarity measurement. In: 2009 International Conference on Computer Technology and Development, vol. 1, pp. 58–62. IEEE (2009)
5. Heusel, M., Ramsauer, H., Unterthiner, T., Nessler, B., Hochreiter, S.: GANs trained by a two time-scale update rule converge to a local Nash equilibrium. arXiv preprint arXiv:1706.08500 (2017)
6. Data World. Tweets dataset for Trump and Hillary (2021)
7. Yang, C., Harkreader, R., Gu, G.: Empirical evaluation and new design for fighting evolving Twitter spammers. IEEE Trans. Inf. Forensics Secur. **8**(8), 1280–1293 (2013)
8. Ferrara, E., Varol, O., Davis, C., Menczer, F., Flammini, A.: The rise of social bots. Commun. ACM **59**(7), 96–104 (2016)
9. 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, pp. 963–972, April 2017
10. Zarei, K., Farahbakhsh, R., Crespi, N., Tyson, G.: Impersonation on Social Media: A Deep Neural Approach to Identify Ingenuine Content. arXiv preprint arXiv:2010.08438 (2020)
11. Floridi, L., Chiriatti, M.: GPT-3: its nature, scope, limits, and consequences. Mind. Mach. **30**(4), 681–694 (2020). https://doi.org/10.1007/s11023-020-09548-1