Tangui Reltgen

Cybersecurity and Cybercrime

Final Paper :
At the dawn of a post-truth era : The threat of Deepfakes on our democratic societies

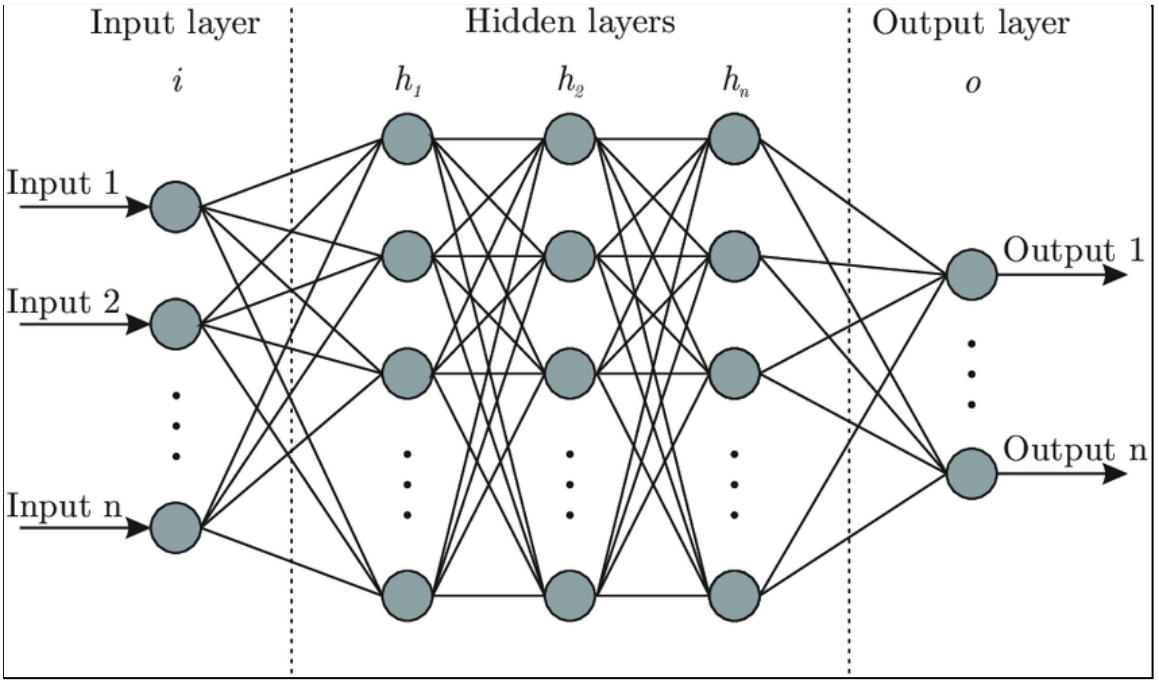
In today's overconnected world, information flows are faster and denser than ever. One can learn in a few seconds what happened thousands of miles away. Social networks have been major actors in this information revolution, by creating a graph-like structure in which any user can share content. Because of the speed and the volume of the flows, it is getting more difficult to distinguish between information and disinformation. Thus, social networks are the perfect place to set a major disinformation campaign, which can be used to manipulate the public opinion or to undermine the political stability of a country. Although disinformation has always been a tool in history, the sheer size of the community of users in major social networks (Facebook, Twitter) and the extreme connectivity of these users makes fake news diffusion a real threat (Fallis, 2020). Even though there is now a relative awareness concerning the spread of fake news, there is a much stronger vulnerability towards fake images or video clips, because people tend to trust those types of information more. Indeed, a new technology of image modification is rapidly emerging, and has a major disinformation potential due to its accessibility and its credibility. This new technology was named "deepfake", a contraction between "deep learning" and "fake". The idea of modifying an image is not new : in 1865, after Lincoln's assassination, many modified pictures were created, sticking his head on other bodies to create convincing lithographies of him. But it is only recently that the technology allowed us to generate extremely realistic videos of people doing or saying things that they never did or said. The first paper dealing with deepfake technology was published in 2016, but it wasn't until the end of 2017 that deepfakes became widespread and raised public awareness after Reddit users used the deepfake technology to put faces of famous actresses on pornographic videos. Since then, deepfakes have been more realistic and easy to generate, with applications such as FaceApp or FakeApp, and open source softwares like TensorFlow and Keras. The progresses made by neural networks are posing a real challenge due to the impossibility of determining if a video is falsified, therefore creating a strong threat to our democracies. In this paper, we will first focus on the technical aspect of deepfakes and on the solutions we have to detect them. We will then propose a framework to study the impact of deepfakes as a disinformation tool by modeling the diffusion of a deepfake video in a social network, using graph theory. We will finally underline the dangers of deepfakes by examining different scenarios in which the deepfake technology is used as a weapon against a country or a company.

I) The technical aspects of deepfakes creation and detection

A) How to create a deepfake ?

a) Neural networks

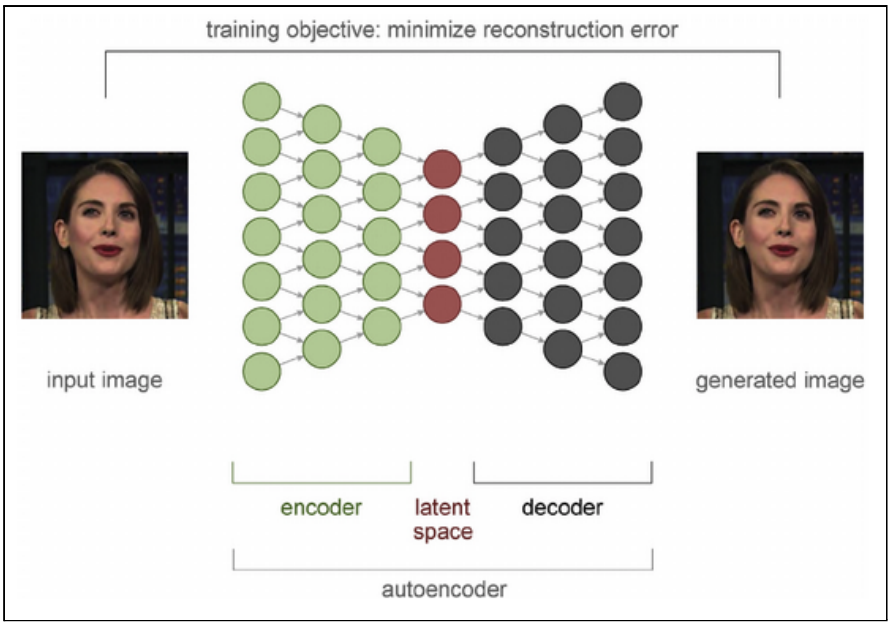
To understand how deepfakes are made, one must first understand the concept of neural networks. A neural network is an interconnected group of nodes, usually organized in layers. Each node is connected to other nodes by one or many edges, can process an input and transmit the resulting output to all the other nodes it is connected to. Each node and each edge has a weight, which is a real number used to increase or decrease the strength of the connexions, and allows it to deal with information in a nonlinear way. The weights are adjusted during the training phase in order to reach the optimal settings, so feedback processes need to take place. Because of that need for training, this technology is often called “deep learning” since the neural network has no prior knowledge on the topic it will be dealing with. An interesting - though not entirely exact - analogy for how neural networks work would be the human brain : nodes are neurons, edges are synapses linking neurons, and the neural plasticity is equivalent to adjusting the weights in the neural network. At the very beginning, all the weights are randomly chosen, so the neural network has to be trained to gain efficiency, thus needing an enormous amount of pictures of the individual one wants to “deepfake”. Because of this important input, celebrities are most often used in deepfakes since there are lots of images of them freely available online.



Simplified graph, showing the three main components of a neural network

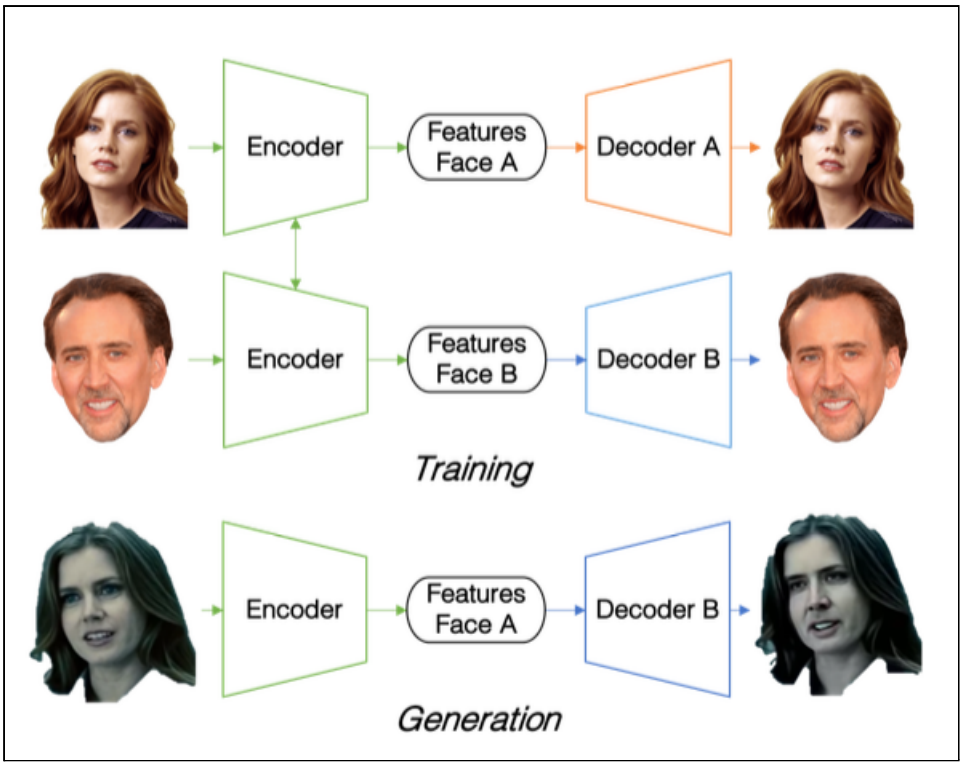
b) Variational autoencoders

Deepfakes can be created by switching the face of a person speaking with the face of another person. For clarity purposes, we will call the person in the original video the “model” and the person used for the deepfake as the “target”. Three steps are necessary : first, extracting the face of the original model in the video, then use it as an input in the neural network to create a matching image with the target face, and finally insert the generated face in the original video. To create deepfakes, one needs to use a specific type of neural network called the autoencoder. The autoencoder is divided into three parts, each with a specific role. First, the encoder takes an important amount of data as input and converts it into a latent space, using dimensionality reduction. The key idea is to reduce the extreme number of characteristics in a human face to a small number of typical facial expressions that will be used as “blueprints” to generate the fake images. These blueprints are stored in the latent space, which is basically compressed information of the input pictures. Finally, the decoder will retransform (decompress) the information mapped in the latent space to reconstruct the facial expressions of the pictures given as inputs.



Anatomy of an autoencoder, taken from Kietzmann, Jan & Lee, Linda & McCarthy, Ian & Kietzmann, Tim. (2019). Deepfakes: Trick or treat?. Business Horizons. 63.

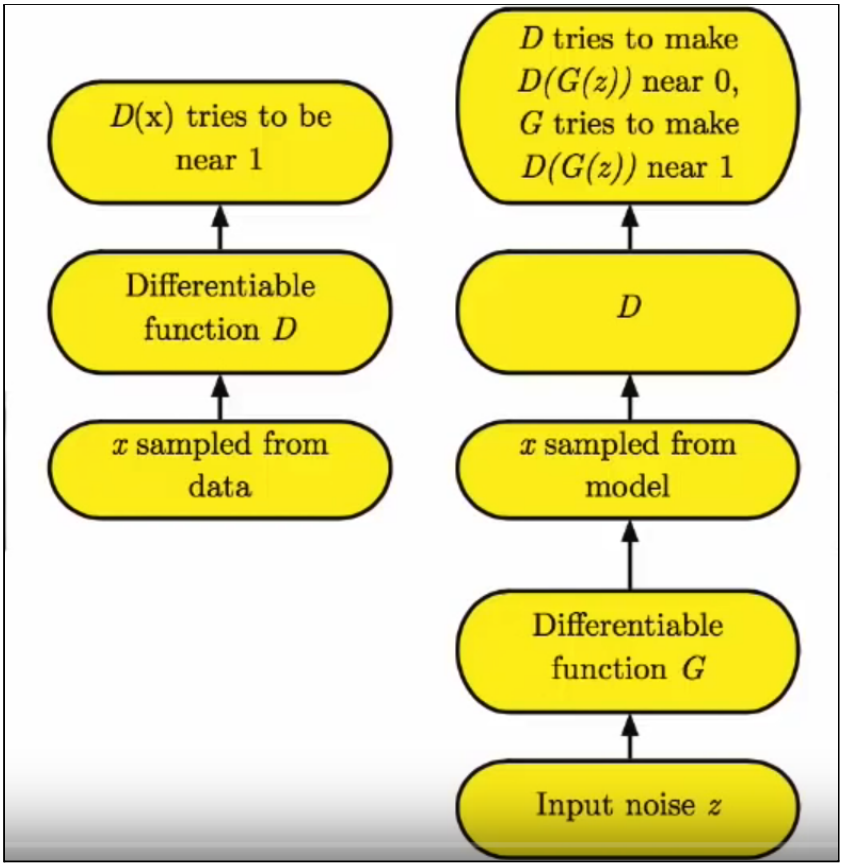
The evaluation process consists of comparing the resemblance between the original images to the output, and to adjust weights if necessary. To create realistic deepfakes, one needs to train two autoencoders (one for the model face and one for the target face), with a shared encoder, but with two distinct decoders. Thus, the resulting latent space will contain significant data for both faces, but when using the target decoder to decompress the data, it will generate the model’s facial expressions on the target face for each frame of the video, which takes lots of time and computational power.



Two autoencoders used to produce a deepfake of the actor Nicolas Cage.
Image taken from Güera & Delp, 2018

c) Generative adversarial networks

Another type of neural network can also be used to create deepfakes, the Generative Adversarial Network (GAN), introduced in 2014. In this type of technology, two neural networks with distinct roles are competing against each other. One is called the “generative” network and the other the “discriminative” network. The generative network tries to generate fake data whereas the discriminative network’s task is to discriminate between real and fake data. They can be metaphorically compared to counterfeiters trying to fool policemen : as policemen learn to detect fake banknotes, counterfeiters produce more and more realistic money. To begin, the discriminative network (D) is trained on a set of data in order to calibrate it (i.e. to maximize the recognition function of pictures from the set of data). Then, the generative network (G) is given a random input (z) and tries to modify it in order to confuse D and make it unable to distinguish between z and x.



Taken from the video : https://www.youtube.com/watch?v=9JpdAg6uMXs&t=1s

We can show using game theory that the outcome of this competition is that the generative network will produce (after a sufficient amount of time) data that is indistinguishable from the sample given to the discriminative network. Using GAN allows us to create extremely realistic deepfakes, but also to generate fake human faces that never existed and that can fool anybody[1].

d) Other emerging technologies

Many other structures of neural networks can be used to produce convincing deepfakes, and the current academic research is blooming in this domain, making it difficult to pinpoint all the new methods emerging. Moreover, the papers are extremely technical and there is a lack of time to assess the efficiency of the new methods. Despite those constraints, we can still evoke two promising technologies, the first being the Variational Autoencoders Generative Adversarial Networks, which are combining the features of both models presented above, and the Swapping Autoencoder in which the image is encoded into two different latent spaces, one that deals with the structure and the other with the texture, thus allowing fine tuning of the image along some parameters. Vocal deepfake technology is also evolving quickly, and it will soon be possible to generate videos with both sound and image falsified in an almost undetectable way. These constantly evolving technologies demonstrate that there

[1] See for instance : https://thispersondoesnotexist.com/

is a constant technological and academic progress in generating deepfakes, and that we should consequently be prepared to face this potential threat, firstly by looking at how deepfakes might be used to create political instability, and secondly by reviewing the detections techniques available to us.

B) The threats posed by deepfakes to our democracies

The current technologies of information diffusion and the low awareness of their users make them prone to large scale fake news diffusion. Indeed, the decentralized structure of social networks, although being an advantage for freedom of speech and plurality of opinions, allows anyone to share information without verifying its integrity. Moreover, due to the high connectivity of users and the popularity of those social networks it is to use them as disinformation or propaganda canals. Far from being marginal, this phenomenon is quite common : Russia and China (among others) are known to employ hackers and “trolls" in order to support their political and military ambitions (Beskow & Carley, 2020). The emergence of large scale data-mining combined with artificial intelligence increases the threat even more. It is now possible to do targeted disinformation on specific users chosen by their political acquaintances or their psychological profile as it was the case during the Cambridge Analytica scandal (Isaak & Hanna, 2018). Due to the high credibility we tend to give to videos, the potential threat of a massive disinformation campaign based on deepfake videos targeting a receptive population can be massive. Moreover, creating a realistic deepfake nowadays is much easier since powerful tools are freely available. Open Source tools such as TensorFlow or Keras can allow a knowledgeable person to generate any kind of falsified video and to potentially use it to harm.

Furthermore, there seems to be a natural tendency among human beings to propagate negative information much faster than positive information : according to the following study, based on 126,000 news stories shared on Twitter, false information was able to spread 10 times faster than true information[2]. Moreover, as shown in various psychological studies, people tend to remember more negative information[3], which underlines the harm potential of deepfakes. So-called filter bubbles can also favor the spread and the credibility of deepfakes, as recommendation algorithms tend to always suggest ideologically uniform content to people on the internet. This reinforces the confirmation bias inherent in human beings, and consequently if a deepfake is designed to target a specific population, it will probably reach an extreme propagation rate in the network, accounting for the fact that many of the “initially targeted community” will share it to prove to others that they were right.

Finally, a major aspect that needs to be considered is the fact that the tools to detect deepfakes are difficult to use, and that many people will become extremely suspicious to “official debunking”, i.e. to governmental claims that a video was altered, especially if this video is involving political personalities or anyone working for the State. The exact opposite could also happen, with a government denying the authenticity of a video/audio file with the claim that it is a deepfake. If large scale deepfake attacks have been previously conducted, such a claim would seem believable, at least to some part of the population.

---

[2] Soroush Vosoughi et al., *The Spread of True and False News Online*, 359 SCIENCE 1146, 1146 (2018), http://science.sciencemag.org/content/359/6380/1146/tab-pdf [https://perma.cc/5U5D- UHPZ].

[3] See*, e.g.*, Elizabeth A. Kensinger, *Negative Emotion Enhances Memory Accuracy: Behavioral and Neuroimaging Evidence*, 16 CURRENT DIRECTIONS IN PSYCHOL. SCI. 213, 217 (2007)

C) How to detect deepfakes ?

A real arm race is currently ongoing between deepfake conceptors and deepfake detectors. The techniques used to generate a deepfake can leave traces on the transformed image, and even if those traces are not detectable by the human eye, they can be detected with the same technology they were created : neural networks. Three main types of inconsistencies can be detected on manipulated videos, and they are all related to the way a deepfake is created. First, since the autoencoder is only trained on the faces, they tend to have difficulties producing images that are coherent with the light and the shades of the original footage. Second, the autoencoder generates a new image for each frame of the video, without considering the preceding or following ones. As a result, the boundaries of the face often become blurred, which can be detected. Finally, this frame-by-frame generation can lead to flickering effects, imperceptible to the human eye but easily detectable by neural networks. D. Güera and E. J. Delp described a method to verify the authenticity of a footage, looking for the flaws we mentioned before. It works with two different neural networks : a convolutional neural network to extract the data in the face-region, and a long short term memory neural network to analyze it. The convolutional neural network (inspired by visual cortex neurons) divides the image in tiles with given properties (like width, height and rgb color), and each tile is given as input to an individual neuron of the input layer, which will then pass it to the convolution (or hidden layer). The output of the CNN is then given as an input to a LSTM (long short-term memory) network, which is a neural network able to process large amounts of continuous data, perfectly suited to handle videoclips. They reach a staggering 94% detection rate of deepfakes in a sample of 600 videos with half of them being pristine. But this paper was published in 2018, and the arms race has not stopped : a year later, P. Korshunov and S. Marcel demonstrated that it was easy to fool two neural networks used in deepfake detection : VGG and FaceNet. Using GAN, they managed to reach between 88.75 and 95% of false acceptance rate (i.e. the deepfake detectors were considering pristine modified videos) on both VGG and FaceNet. These two papers illustrate well the constantly ongoing competition towards more sophisticated techniques of creation and detection of deepfakes. Despite this, it seems that new methods of detection are emerging, which might be able to give an edge to deepfake detection. S. Agarwal et al. argue for the ability to identify specific facial expressions of a well known political personality using GAN. These facial features would then serve as a signature to authentify videos, being reliable enough to detect both face-swap (93% of accuracy), lip-sync (95%) and puppet master (94%) types of deepfake, but also impersonation by comic actors (94%) on 10 seconds clips, which is remarkable. But we should not fool ourselves : as offensive research is probably done by major states, we have little information about how sophisticated the deepfake technology can get, and therefore we have no reason to think that deepfake detection can reach a perfect result.

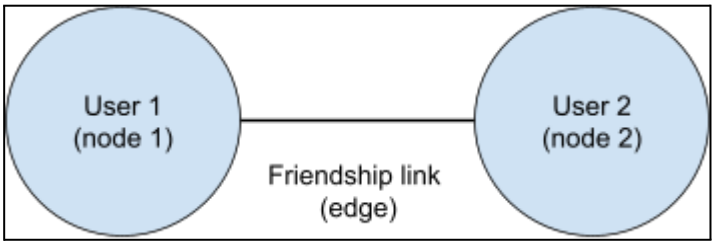
II) Modelisation of a deepfake propagation

Modeling the diffusion of a deepfake in a social network allows for a deeper understanding of the mechanics of social contamination, and thus gives opportunities to study

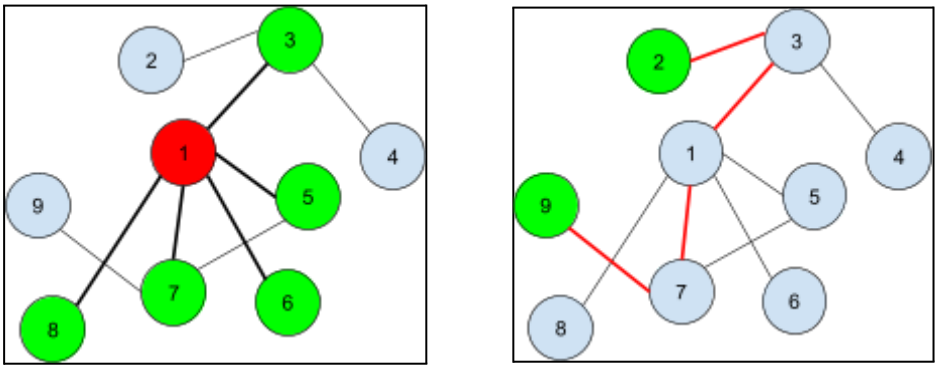
counter-measures to limit the deepfake threat. We chose to create a simulation with a graphic interface to show the dynamics involved in deepfake “propagation”. By propagation we mean the increasing number of people who will be in contact with the modified footage as it is shared by more and more users of the network. Having no prior knowledge in coding whatsoever, creating such a program was a real challenge. We used Javascript language for coding, D3js library for the visualization, and graph theory to build a realistic social network. We learnt Javascript basics using W3Schools tutorials and freeCodeCamp Youtube videos. StackOverflow users were of great help to help us solve the (many) technical problems we encountered.

A) Building a realistic social network

The first challenge to simulate the propagation of a deepfake is to reproduce the main characteristics of a social network. Many social networks are today used to share information, but we chose Facebook as a model, since the friendship links between users are a good approximation of the channels by which the deepfake is going to propagate. Moreover, Facebook has billions of users and has been used multiple times to share information during large political and social events, so it would be a good entry point for a malicious offender. The best way to analyze a network is to use graph theory, which provides powerful tools to understand its structure. Thus, in our model, each node will represent a user and each edge will be a friendship link between two users.

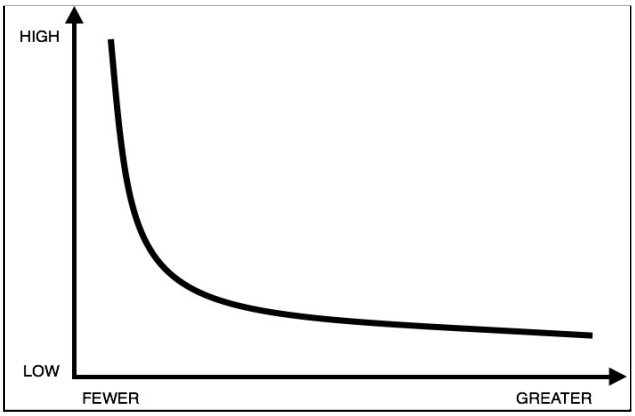


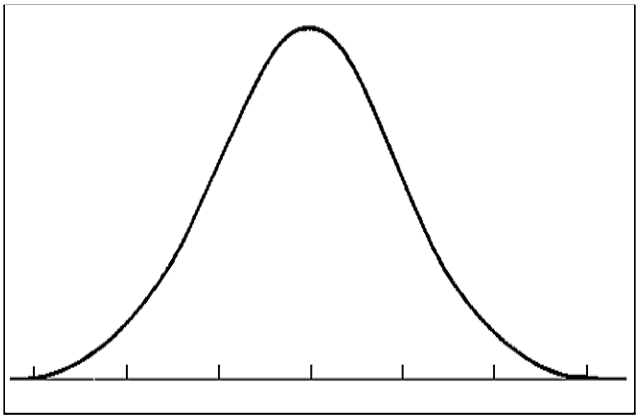
Some useful properties of a graph are :
- the degree of its nodes (the number of neighbors a given node has)
- the distance between two nodes (the minimum number of edges needed to reach a target node from a source node)



Left : degree(1) = 5. Right : distance (2, 9) = 4

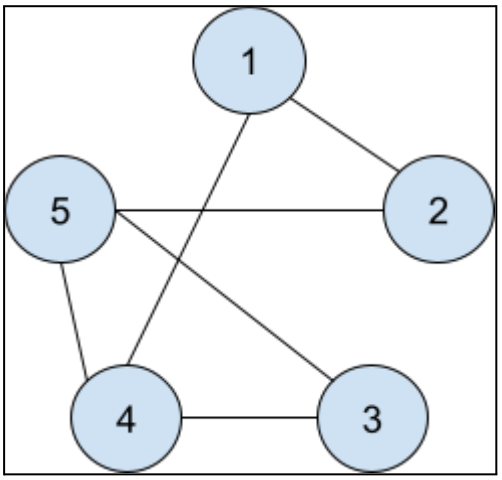
- the distribution of the degree of the nodes (what statistical distribution do they follow ?)

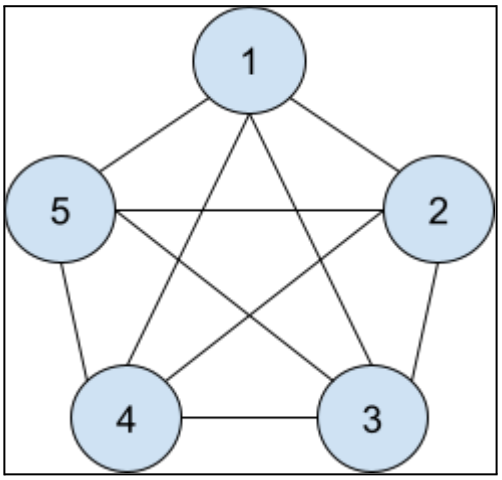




Left : Power law distribution, often found in real world networks. Right : Normal distribution found in random networks.

- the density/sparsity of the graph (the total number of edges in the graph divided by the maximum number of edges possible, which is $E(max) = \frac{n(n-1)}{2}$ where n is the number of nodes). A perfectly dense graph is called a clique.

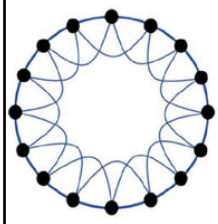
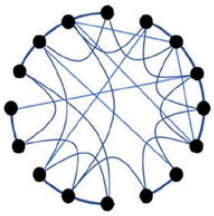
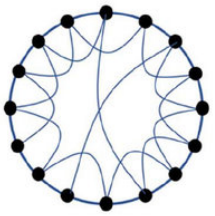
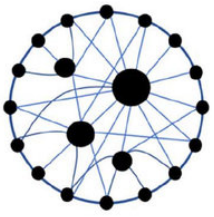




Left : a sparse graph, Right : a 5-clique

The theory of “six degrees of separation” developed by the Hungarian Frigyes Karinthy in 1929 states that two people randomly chosen anywhere in the world are connected by at most 6 people. In the language of graph theory, this means that the maximum distance in the graph of all the social relationships in the world is six (of course, in practice it cannot be true because of small isolated communities, i.e closed subgraphs). Those types of graphs are deemed “small world networks”, meaning that the average distance between two nodes is low. According to a 2016 Facebook research, the average distance between two users of the

application is 3.57, confirming that a Facebook-like graph needs to have a small average distance[4]. A naive approach (which was our first attempts) would be to create a random graph by giving to each node an index and then generating the list of all the possible pairs of nodes (excluding the “self-linked” nodes and keeping only the lower part of the adjacency matrix since our graph is undirected and thus 2,3 describes the same edge as 3,2), and then to randomly pick a given number of pairs to create the edges. But this creates graphs with a higher average distance than small-world networks. Moreover, using the published literature on the Twitter social graph, we learned that another of its characteristics was that it is a scale free network, meaning that the degree of its nodes follow a power law distribution, and that it has a higher connectivity and density than a random graph. This is somewhat intuitive : there is a limited number of very well connected nodes (users with millions of followers) and a much higher number of moderately connected nodes.

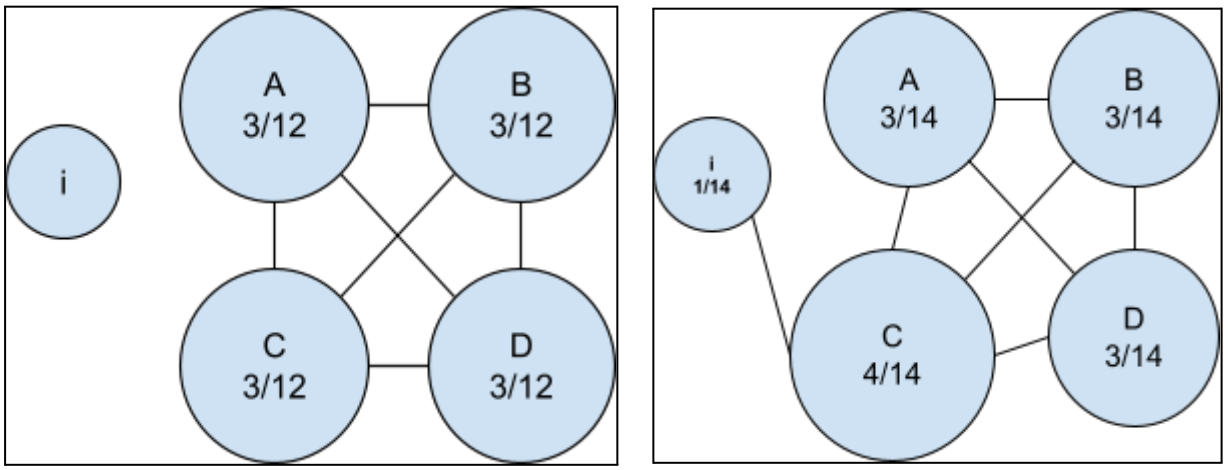
Network type	 Regular Lattice	 Random	 Small World	 Scale-free
Average distance	High	Medium	Very Low	Low / Medium
Degree distribution / hub effect	Linear / None	Normal Law / Small	Log-Normal Law / Moderate	Power Law / Very High
Clustering coefficient	High	Low	High	Medium

Four different types of networks with varying characteristics. The regularity of the lattice graph immediately eliminates it as a realistic candidate. The random graph seems appropriate at first, but follows a normal degree-distribution which is not what is observed. We had to find a compromise between a small-world model and a scale-free model, in order to have both a low average distance and a very high hub effect.

The Barabasi-Albert model is well suited for our simulation. It generates scale free networks using a preferential attachment mechanism. It works as follows : we start from a clique of size m0. We then add (N - m0) nodes successively. At each step, the probability p(k) that a new node k attaches to an already existing node i is p(k) = degree (i) / sum of the degrees of all the nodes in the graph. At the beginning, therefore, all the nodes of the clique have the

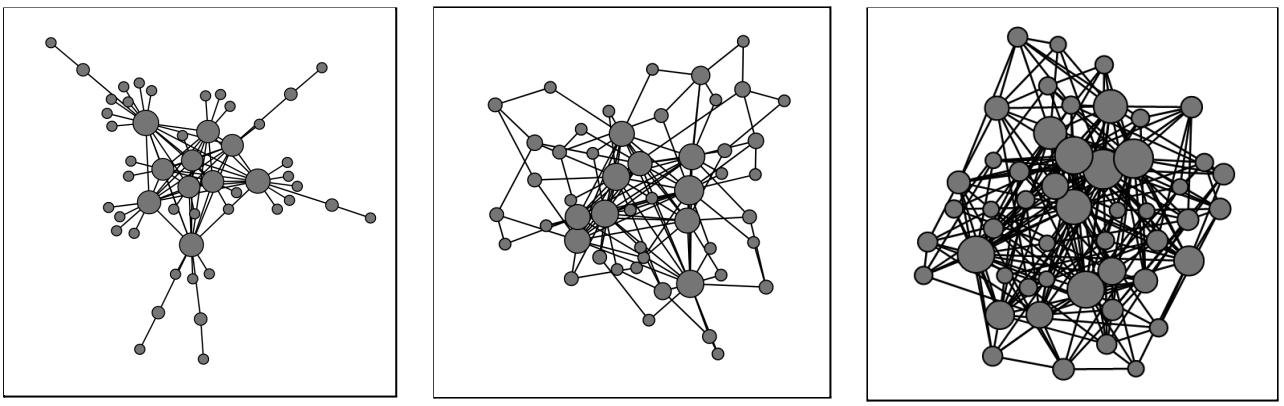
[4] https://research.fb.com/blog/2016/02/three-and-a-half-degrees-of-separation/

same probability of attracting a new node. However, a small imbalance is enough to cause a snowball effect that benefits the nodes with a high degree.



Left : Step 1, each node of the initial 4-clique has 4/12 chances to attract the entering node i. Right : Step 2, node i is connected to node C, thus increasing the degree of node C and its probability of attracting a new node. Note that the total degree of the graph will always be twice the number of edges.

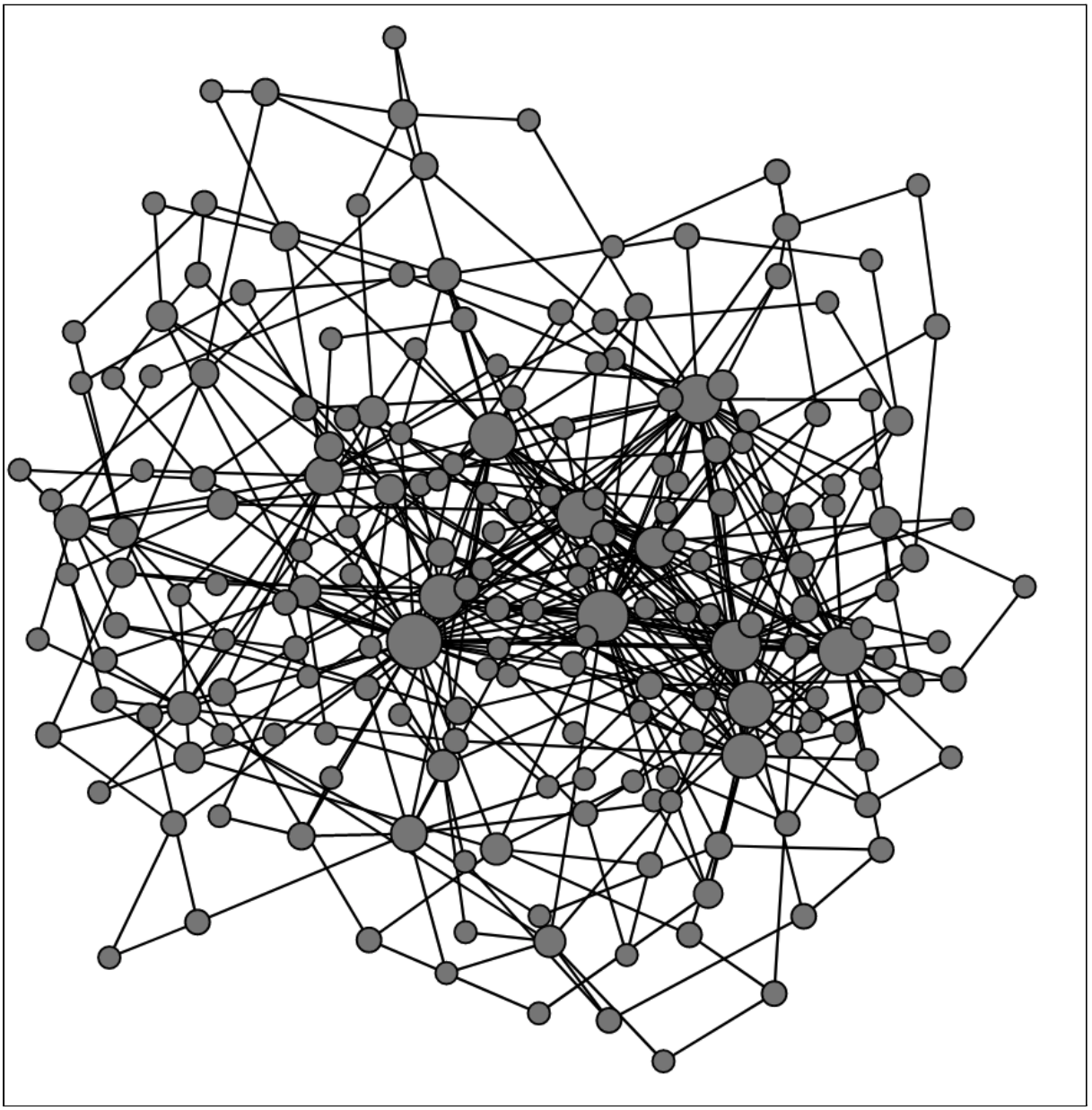
The model contains an attachment parameter M which defines the number of links that an incoming node will create. The most realistic social-like network can be obtained with M between 2 and 5. The higher M, the denser the graph will be, and the shorter the distance between two randomly chosen nodes.



From left to right : fixed N = 50, fixed m0 = 10, variations of M = 1, 2, 5

Finally, it is important to note that each graph created will be different due to the random nature of the attachment process. The question of graph visualization also arises: how to make a graph of a considerable size readable? The program uses the d3js library which contains a force-directed graph functionality. This allows to shape the spatial structure of the network to facilitate graphic rendering. For more ease, it is possible to move the nodes with the mouse (click and drag), because due to their proximity, two nodes can sometimes give

the illusion of being linked when they are not. The size of the nodes is also proportional to their degree, for a better understanding of the graph.



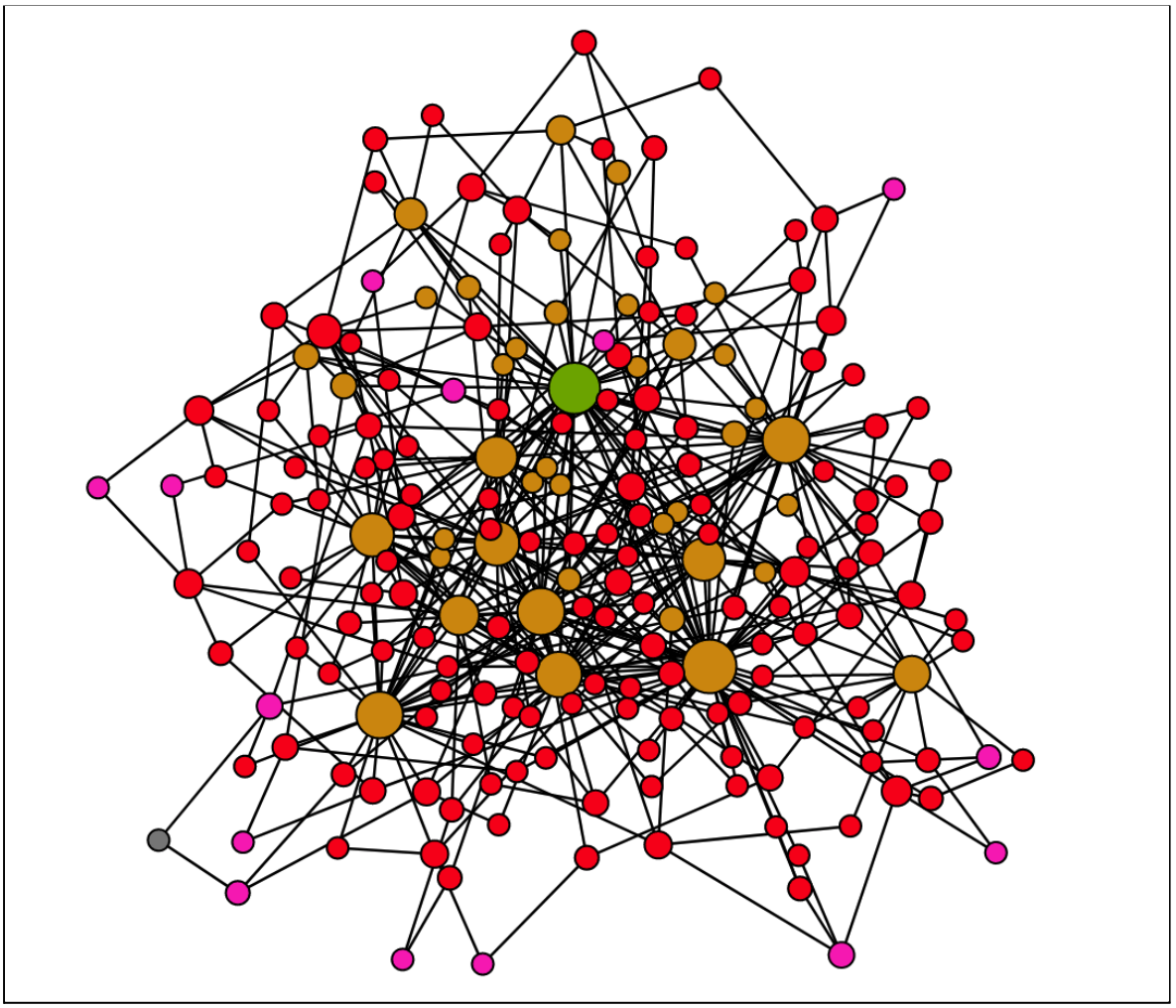
With m0 = 11, N = 200 and M = 2 we get the following figure

We can observe that there are a few nodes with a high number of connexions (mostly the ones from the initial clique) and that most of the nodes have a low degree.

B) Propagation mechanism

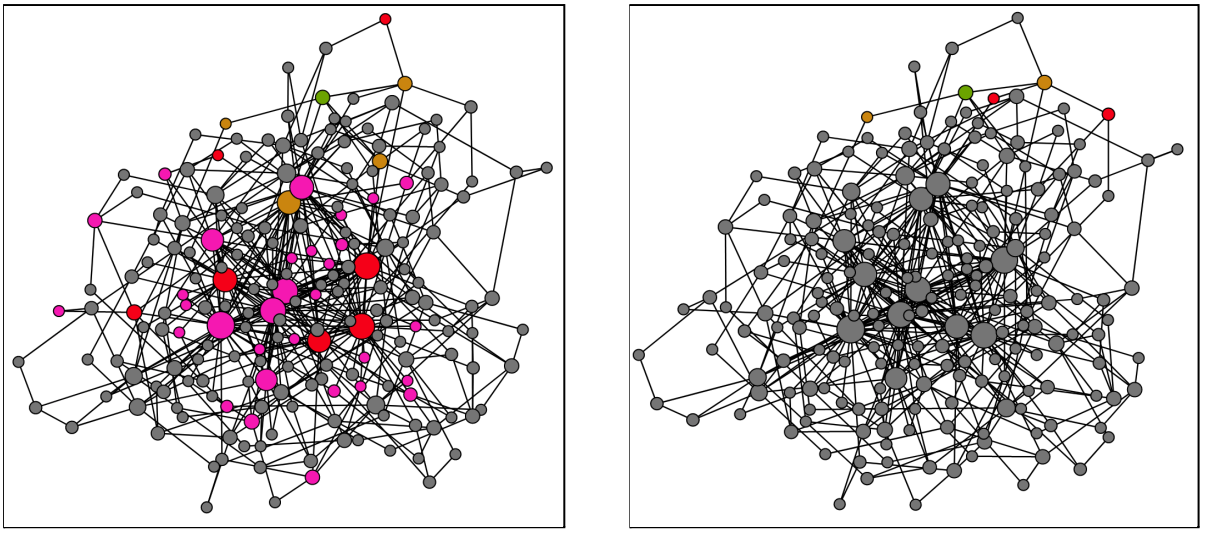
Once the parameters of the graph have been chosen, two other parameters must be chosen to determine the course of the simulation. The first parameter is the contamination factor, (i.e. how believable the deepfake is, and how likely people who have seen it will share it). Here, it is a constant between 0 and 1 that determines the probability that a contaminated

node will contaminate its direct neighbors. Thus, for a value of 1, all the direct neighbors of the node will be affected. The other parameter to be defined is the number of iterations of the simulation. Indeed, the propagation of fake news occurs over time, and some people can see the deepfake multiple times before sharing it. The number of iterations represents the number of times someone is exposed to the deepfake. At each iteration, the program looks at the neighbors of all contaminated nodes, (who can potentially be contaminated) and then adds them according to the chosen probability to the list of new contaminated nodes, and so on. To distinguish when a node has been contaminated, a color code is determined: the node at t+0 is green, orange at t+1, red at t+2, magenta at t+3, indigo at t+4, etc...
To clarify things, let's give an example : for a contamination probability of 1 and a number of iterations of 3, it is highly probable that the majority of the nodes in the graph are contaminated. Indeed, this means that all nodes located at a distance of 3 or less from node zero will be contaminated.



Example of quick contamination in a graph with m0 = 11, N = 200 and M = 2. The contamination parameter is set to 1, and the number of iterations to 3. We observe that most of the nodes are contaminated at iteration 2 (red color) and that all the nodes but one (in the lower left corner, in grey) are infected by iteration 3.

Let us specify that for lower values, the variability of the contagion may be very large: since the future contaminated nodes are randomly chosen among the neighbors of node zero, if by "chance" nodes with a very large degree of contamination are found, then the number of potential cases will increase.

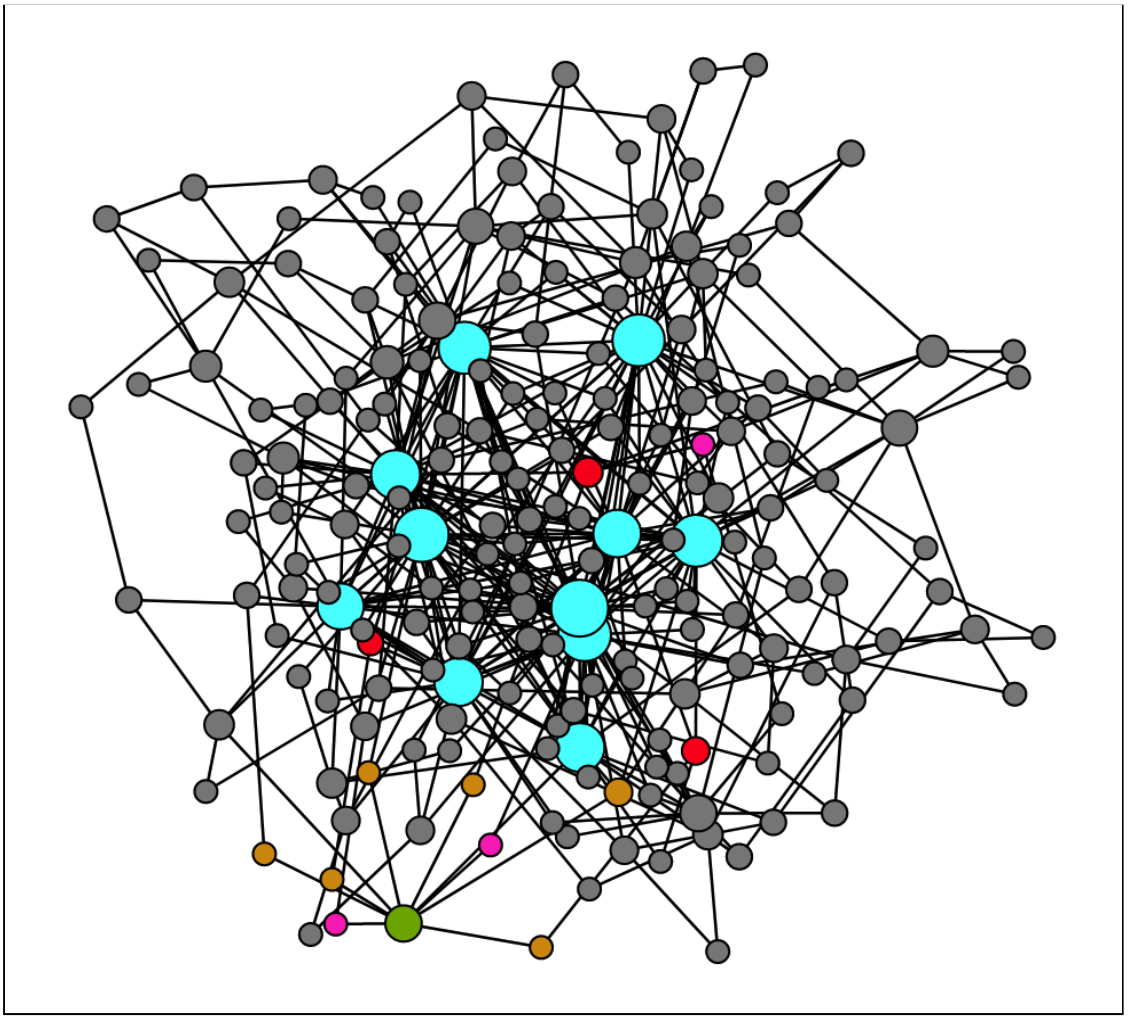


Two contamination processes in the exact same graph, (m0 = 11, N = 200 and M = 2) with a probability of contamination set to 0.3 and the number of iterations to 3. Though the exact same entry point is used, It is clear that on the left graph the deepfake was more successful in its propagation than on the right side (where no nodes were contaminated at iteration 3). The difference is easy to explain : one can see that on the left graph a major node was reached (by chance) at iteration one (orange color), leading to the contamination of 4 hubs at iteration 2 (red color), thus allowing for a deep propagation at the final iteration. On the contrary, on the right graph no important node was reached, making it hard for the deepfake to spread efficiently.

To trigger the propagation, one must double click on the node that one wishes to be the entry point. The degree of the entry point plays a fundamental role in the spread of the deepfake : if it is very high (i.e. a celebrity or someone that is highly followed on the social network), even a low contamination factor (low believability) will not be enough to contain the propagation. Conversely, a node at the periphery will not allow fast diffusion in the graph. The number of iterations chosen should not exceed 5 due to the exponential computation time of the simulation. A delay of a few seconds is possible for large graphs and high values. For a deeper understanding of the contagion mechanism, it is possible to use the web console (Ctrl/Cmd + Alt + K): the identification of infected nodes and their infection iteration is displayed. (Note: a node can be infected at iteration 1 and then 3, in this case its color is the same as the first iteration of infection). All nodes have an identifier that can be seen by hovering the mouse over them. The model enables us to have a visual representation of the dynamics behind the propagation fake news (embodied by a realistic deepfake), and underlines three major factors :

- the importance of the graph structure, which allows quick propagation because of the low average distance between two nodes.
- the role of random mechanisms which lead to high variability in the final state
- the role of "hubs" i.e. nodes with high connectivity which are key actors of the propagation

To study the possible countermeasures to prevent deepfake propagation, we added a mechanism that allows us to immune some nodes of the graph, making them impossible to become infected. This action models a good prevention policy implemented by states and non-governmental actors, or so called “debunking” operations in which the falsehood of a deepfake is revealed. The (inelegant) trick we used to do so was to increase the ID of a circle by 500 if it is immunized, and to forbid the contamination function to deal with IDs higher or equal to 500. The safe nodes are displayed in cyan color. (The button “START IMMUNIZATION” must be enabled before clicking on the nodes, and the button “STOP IMMUNIZATION” must be clicked before starting the contamination).



Example of useful immunization (m0 = 11, N = 200 and M = 2, i = 3) : by selecting the 11 most connected nodes (from the original clique) and making them immune, we strongly reduce the diffusion of the deepfake in the network, despite an originally high contamination parameter (z = 0.6).

This simulated model allows us to better understand the dynamics behind fake news diffusion, and more specifically deepfake diffusion. We showed that by successfully immunizing a small number of key nodes (major accounts that are trusted/frequented by everyone), it is possible to drastically restrain the propagation of the deepfake. But our simplistic model can be improved in order to be closer to real-life events.

C) Improvements of the model and real-life comparisons

Due to the lack of time and coding knowledge, we kept our model as simple as possible. Thus, it may not entirely grasp the complexity of social contagion phenomena on a social network. We propose further improvements that could be made in order to make it more realistic. Three specific upgrades of the model can be identified, based on the current flaws it has.

First, the program only allows one entry point to spread the deepfake, which is not really consistent with what has been observed in recent online disinformation campaigns. Usually, many automated accounts (so called bots) are used, backed by humans operating false accounts. Thus, dealing with multiple entry points would be a great upgrade and would better model a deepfake attack scenario. It would also allow us to compare the efficiency of a strategy based on multiple weakly connected nodes to a strategy targeting a hub of the network.

Second, the propagation mechanism is somewhat naive because it is very linear : in our model every node transmits the deepfake with the same believability. However, this is not the case on social networks, where bigger accounts have a much greater “believability potential” than the smaller ones. To model this, we could make the infection parameter of a node (the probability it infects its neighbours) proportional to the degree of the node. Consequently, infecting a hub would be even more beneficial for the attacker since it would guarantee him a deeper penetration in the network. Another interesting improvement of the infection parameter would be to make every node "aware" of its surroundings : the infection parameter would increase in a nonlinear way (using a sigmoid activation function like in neural networks) depending on the proportion of its neighbours that are infected (e.g. a slow increase while less than half of the neighbors are infected, but a much steeper increase once this activation threshold is reached). This mechanism would account for the conformist pressure shown by the famous experiment of psychologist Solomon Asch.
Thirdly, an improvement of the immunization processes would be beneficial to make the model more realistic. Instead of completely stopping contamination in a passive way, the immune nodes could reduce it, acting like infected nodes and "sending" immunization messages to their neighbors in order to reduce their probability of being contaminated. This would model well debunking campaigns which diffuse in the network exactly like fake news.
All these improvements could allow the program to be closer from reality, and therefore to develop prevention tools in order to be prepared against deepfake attacks and disinformation campaigns. Of course, it will increase the computational complexity of the program, but with some optimization and parallelisation it could work properly.

III) Two scenarios involving a deepfake attack by a state-like actor

In this section we present two potential scenarios of attacks using deep fakes.

A) Undermining the political stability of a country

State inference in electoral times is well documented and may already have had drastic consequences. Indeed, the 2016 American elections and the Brexit vote were likely influenced by online disinformation campaigns probably orchestrated by Russia (Narayanan et *al*., 2017). These campaigns based their strategy not on the believability of the false information they spread, but rather on targeting receptive individuals using their online data to map their psychological profiles. A strategy combining on the one hand a targeting approach and on the other strongly believable fake news (using deepfakes) would be devastating. We propose a scenario in which a malevolent actor with state-like ressources would undermine the social and political stability of a country during election time. The attacker would need to choose a country in which citizens already have a very low confidence in politics. According to the OECD, in 2019 only 45% of the citizens were trusting their government[5], a strong sign of growing political mistrust which is the perfect environment for deepfake attacks, since governmental debunking campaigns would then be dismissed by the suspicious population. France seems to be a perfect target for a deepfake attack since it combines the following characteristics:

- a climate of strong social tensions (embodied by the yellow vest movements which led to violent riots all other the country),
- a complete lack of trust in the government and the President (in a recent poll asking French citizens to rate their feelings towards the government from 0 (none) to 10 (extreme), 55% answered 6 or more for "anger" and 12% answered 10)[6]
- An incoming electoral period, with presidential elections to be held in April 2022

Consequently, France could be the perfect target for a deepfake attack aiming to trigger violent social unrest. Such an attack could take advantage of how the French president Emmanuel Macron is perceived by the population : many people consider him too arrogant and authoritarian with a king-like attitude towards French citizens[7]. Thus, the malevolent attacker could proceed in three steps
First, by creating months before the presidential election thousands of fake Twitter accounts and making them look realistic by sharing consensual content (such as memes, verified news, science vulgarisation etc.). Such a strategy should allow these accounts to gain at least a small number of "real" followers, which is crucial to succeed. The same process could be done with Facebook accounts, but with political publications from the far-right to the far-left of the spectrum in order to increase the breadth of the target : it would be optimal to

[5] https://www.oecd.org/gov/trust-in-government.htm
[6] https://www.sciencespo.fr/cevipof/fr/content/les-resultats-par-vague.html (see "Vague 12bis)
[7] https://www.publicsenat.fr/article/politique/un-an-de-macron-c-est-une-vraie-logique-jupiterienne-fustige-pascal-pavageau-86101

infiltrate large Facebook groups which are openly against Macron's party (reaching hubs of diffusion), but also non-political groups. Finally, the attacker would need to create a small number of YouTube accounts publishing videos on popular subjects like gaming or funny video compilation, (but without ever showing a face and using text-to-speech generators in order to have a realistic voice).
The second phase of the attack would be the creation of the deepfakes. We propose a scenario in which Macron would be filmed during a meeting with its advisors saying that he would stay in power whatever the results of the elections are. The angle of the camera should lead the spectators to think that the shot was secretly taken by someone in the room, in order to make it more believable. The quality of the video and the audio deepfake is essential, but we can suppose that an actor with state means will not have difficulties to generate extremely realistic deepfakes.
Finally, the last phase of the attack would be the diffusion of the video in social networks. The attacker would use all the fake Twitter accounts to propagate the video, using multiple entry points in order to reach a hub, thus guaranteeing fast propagation. The speed of diffusion in the network is crucial for the success of the attack : the deepfake will only have a short window of time before it gets debunked and denied by the government. However, if the snowball effect of sharings and retweets is fast enough, then debunkings and denials will have the opposite effect and will act like a confirmation that the video is genuine. If both Facebook and Twitter accounts are involved in the operation, the chance of triggering such a snowball effect is quite high. The YouTube channels could be used as a bait to reinforce the belief that there is governmental censorship in action to try to delete the video: the Twitter accounts would share the links of the channels with the original sequence, but in the meantime the channels would be shut down by the attacker, reinforcing the feeling that the government is doing everything to prevent this video from getting a large audience. Moreover, the attacker could flood the social networks with regular fake-news (i.e. non-deepfake) in order to overload debunking agencies and to increase the fears and the uncertainty of the population. In a context of strong social tensions, we believe that this would trigger massive riots and uprising in the main urban centers, enough to destabilize the country for several days and to change the course of the election. We chose to present this scenario to expose the weaknesses of our democracies towards fake news propagation, a threat greatly enhanced by the strong believability of deepfakes. There is therefore a need to engage in prevention campaigns in order to raise public awareness and limit the vulnerability created by social networks.

B) Manipulating the stock market

The deepfake technology could also be used to manipulate the stock market, either by a company willing to destroy a concurrent or by cybergroups looking for ways to earn millions of dollars by betting on the price of an action. Fake news already have shown that they can strongly impact the stock market : on the first of December 2017, the information channel ABC News reported that Trump had directed Michael Flynn to contact Russian officials during the 2016 campaign, which led to an immediate fall of the Dow Jones by 38 points in half an hour (equivalent to a 341 billion loss, reduced to 51 billion after the clarification by ABC News)[8]. More recently, Elon Musk's declarations about Bitcoin and DogeCoin (although not fake news) have led to extreme volatility of their prices, comforting our thesis that if one

[8] https://www.institutionalinvestor.com/article/b1j2ttw22xf7n6/Fake-News-Creates-Real-Losses

tweet is enough to create market perturbations, a realistic video would have even greater impacts. These impressive events allow us to imagine a scenario where a deepfake of a major CEO giving false information about his company (either positive like a new investment/product or negative such as financial issues or lawsuits) would be used to collect huge profits by buying/selling the company's action at the right timing. Such an attack could have devastating consequences and even be the trigger for a bigger collapse of the market if the economic situation is already unfavorable. Therefore, deepfakes are not just a public policy issue, and companies should be involved in debunking campaigns in order to maintain the trust of their customers and to protect their image and reputation on social networks.

In this paper, we focused on the threat that deepfakes represents for our societies. We first explained how deepfakes are created using different types of neural networks and how deepfake detection can be achieved, but we showed that the constant evolution will lead to deepfakes that are harder and harder to expose. Then, we built a model to show how deepfakes could propagate in social networks and we underlined the importance of hubs in the diffusion process. We also pointed out that a good prevention campaign against deepfakes might be helpful to limit social contagion, especially if well targeted. Finally, we imagined two different scenarios of a deepfake attack in order to realise how vulnerable we are today and how crucial it is to take measures against deepfakes, since they will be part of the cyber-arsenal in a very close future.

N.B : For the program code, you can contact me at tangui.reltgen@sciencespo.fr

Bibliography :

- Agarwal, S., Farid, H., Gu, Y., He, M., Nagano, K., & Li, H. (2019). Protecting World Leaders Against Deep Fakes. *CVPR Workshops*.

- Beskow, D. & Carley, K. (2020). Characterization and Comparison of Russian and Chinese Disinformation Campaigns

- Campbell, C., Plangger, K., Sands, S., & Kietzmann J. (2021) Preparing for an Era of Deepfakes and AI-Generated Ads: A Framework for Understanding Responses to Manipulated Advertising, Journal of Advertising

- Citron, D. K. & Chesney, R., Deep Fakes: A Looming Challenge for Privacy, Democracy, and National Security, 107 California Law Review 1753 (2019).

- Fallis, D. The Epistemic Threat of Deepfakes. *Philos. Technol.* (2020).

- Güera, D. & Delp, E. J. "Deepfake Video Detection Using Recurrent Neural Networks," 2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), 2018, pp. 1-6

- Isaak, J. and Hanna, M. J. "User Data Privacy: Facebook, Cambridge Analytica, and Privacy Protection," in *Computer*, vol. 51, no. 8, pp. 56-59, August 2018

- Kietzmann, Jan & Lee, Linda & McCarthy, Ian & Kietzmann, Tim. (2019). Deepfakes: Trick or treat?. Business Horizons. 63.

- Kirchengast, T. (2020) Deepfakes and image manipulation: criminalisation and control, Information & Communications Technology Law, 29:3, 308-323

- Korshunov, Pavel & Marcel, Sébastien. (2019). Vulnerability assessment and detection of Deepfake videos. 1-6.

- Narayanan, V., Howard, P. N., Kollanyi, B., & Elswah, M. (2017). Russian involvement and junk news during Brexit. *The computational propaganda project. Algorithms, automation and digital politics.*

- Park, T., Zhu, J., Wang, O., Lu, J., Shechtman, E., Efros, A.A., & Zhang, R. (2020). Swapping Autoencoder for Deep Image Manipulation. ArXiv, abs/2007.00653.

- Rössler, Andreas & Cozzolino, Davide & Verdoliva, Luisa & Riess, Christian & Thies, Justus & Nießner, Matthias. (2019). FaceForensics++: Learning to Detect Manipulated Facial Images.

- Verdoliva, L. (2020). Media Forensics and DeepFakes: An Overview. IEEE Journal of Selected Topics in Signal Processing. PP. 1-1.

- Westerlund, M. 2019. The Emergence of Deepfake Technology: A Review. *Technology Innovation Management Review*, 9(11): 40-53.

- Whyte, C. (2020) Deepfake news: AI-enabled disinformation as a multi-level public policy challenge, Journal of Cyber Policy, 5:2, 199-217

Websites :

- Deepfakes: Face synthesis with GANs and Autoencoders : https://theaisummer.com/deepfakes/
- Experts fear face swapping tech could start an international showdown : https://theoutline.com/post/3179/deepfake-videos-are-freaking-experts-out?zd=1&zi=hbmf4svs
- What The Heck Are VAE-GANs? : https://towardsdatascience.com/what-the-heck-are-vae-gans-17b86023588a
- Latest Model That Might Replace GANs To Create Deepfakes : https://analyticsindiamag.com/latest-model-that-might-replace-gans-to-create-deepfakes/
- GANs vs. Autoencoders: Comparison of Deep Generative Models : https://towardsdatascience.com/gans-vs-autoencoders-comparison-of-deep-generative-models-985cf15936ea
- Deep Fakes: A Looming Crisis for National Security, Democracy and Privacy? : https://perma.cc/L6B5-DGNR
- Deepfakes and the world of Generative Adversarial Networks : https://medium.com/@lennartfr/deepfakes-and-the-world-of-generative-adversarial-networks-bf6937e70637
- Understanding Latent Space in Machine Learning : https://towardsdatascience.com/understanding-latent-space-in-machine-learning-de5a7c687d8d
- Three and a half degrees of separation : https://research.fb.com/blog/2016/02/three-and-a-half-degrees-of-separation/

Youtube Video :

- https://www.youtube.com/watch?v=9JpdAg6uMXs&t=1s

Other Resources used :

- D3js library : https://d3js.org
- Stackoverflow : https://stackoverflow.com