University of Padua

Interdishiplinary Project 4

Fake News in Spain:
Impact of Fake News during Spanish Elections of November 2019

Work Group:
Paula Abraham
Neus Frau
Giacomo Loreggia
Simone Favaro

Supervisors:
Tomaso Erseghe
Caterina Suitner
Leonardo Badia

10 February 2020
AY 2019/2020

Contents

Chapter 1

Introduction

Nowadays social media has an important role in the daily lives of people around the world. They have turned out to be a revolution, they have been able to create jobs that 10 years ago nobody would have imagined. They have changed the concept of the way of life. At this point, it is not absurd to think that social networks can influence the results of the elections of the president of a country. We are referring to the famous fake news, a form of journalism that aims to persuade its readers with totally or partially false information, to change their opinion or judgment on a certain subject. In the case of fake news in politics, it has been seen in several cases. The American elections in 2016 is probably the best known and best example, of which has been said that, if fake news would not have been circulating through social networks in favour of Donald Trump, he would not have won the elections.

Thanks to the rapid evolution of social networks, they have become the propitious site for the dissemination of news, and among them, of course, fake news.

In Spain, the voters lack of information plays an important role when it comes to the power of fake news. Spaniards trust whats been spread in social media and TV without doing their own research, which leads to some political parties that have more power over spreading channels to take control and influence what is been said about them and their opponents.

In 2019, spanish politics were found in a difficult situation due to the rise of extreme right - winged parties that made their way to the top using a conservative, liberal and racist speech that involved the misunderstanding and the manipulation of information, catching the attention of the least informed voters. This, in addition to the also growth of young left and central - winged parties, influenced in the fracturation of the votes in the elections realized in April, which resulted in no political party obtaining the absolute majority and forcing the government to summon another poll in november of the same year.

Twitter, which is a massive social network that everyone has access to, nowadays, can be used as a tool to manipulate, in this specific case, the voters view about a political party easily. Anyone with a considered amount of followers

and an strong opinion towards a certain politician can use the platform to divulge a fake new that patronizes his political point of view. Hopefully, due to the raising of this phenomenon, fact - checkers and journalistic tools dedicated exclusively to revise facts and news in order to avoid the spreading of fake information appeared to help the communities have access to veracious information. In this work, we analyzed which role played Twitter in the elections convocated in november 2019, regarding the spread of fake news, analyzing the accounts geolocation to see if they have any impact in the autonomous community they live in regarding the amount of votes a party got, and the accounts communities to see if a political party is more correlated to a fake news provider.

Chapter 2

Dataset

Our goal has been retrive tweets from *politicians*, *news providers* and *fake news providers*[1], from the ten days before Spain elections of November 2019 to the day of the actual elections. We gathered manually 159[2] Twitter accounts, categorized in the following sense:

- 99 most influencing politicians (`pol`), 20 for each one of the 5 biggest parties (*Psoe*, *PP*, *Vox*, *Unides Podemos*, *Cs*) [1];
- 40 news providers (`news`), from the most influencial news outlets in Spain [2];
- 20 fake news providers (`fnp`), found both in [3] and [4].

For each one of their accounts, we got all the tweets (both *statuses* and *replies*) that had been posted during the analysis time frame, using the Twitter API [5] [6]. At the same time, starting from that set of tweets, we got all the accounts that were mentioned in them, their geographic location and the accounts that retweeted them.
At that point of the dataset creation, we had to confront two problems that were not fixable: the first one is that not all accounts had their *geolocation data*: for this matter, we had to manually find some of them, and still for some of the fake news accounts we were not able to find the data. The second one was that we only could access the last 100 accounts who retweeted a certain tweet. As a result, we could not get all the retweet data for the tweets which had more than 100 retweets.

After collecting all the data, the raw dataset we got had a total of about 33k tweets, 360k retweets, 13k mentions. In order to create a more manageble one, we had sieved lots of links, keeping only the mentioned and retweeting accounts which belonged to our initial list of 159 accounts (we just set the domain and codomain equal to the initial users we started from).
In our analysis, we ended up using the following two types of links:

[1] We will use the abbreviation *fnp* for the rest of the review.
[2] They used to be one more, but the account ended up be not avaiable.

- the *mention links*, that is A mentions B only if A published at least one tweet in which he mentioned B;
- the *retweet links*, that is A retweets B only if B is on the list of accounts that retweeted at least one tweet from A.

Finally, we merged the the mention and the retweet links datasets, giving them the same weight. We noted that both dataset had directed links, but the direction of mention links did not match the retweet links one. In fact, mentions mapped from mentioning users to users that are mentioned; on the other hand, retweets mapped from users that are retweeted to retweeting users. We fixed that inverting just the direction of the retweet links.
The final dataset that we actually used for almost all the analysis contained 5531 links.

Chapter 3

Analysis and Results

3.1 Geolocation Analysis

Thanks to the *geolocation analysis*, we were able to represent geographically the retweet and mentions links between users. We could also find information about what regions got more people interested in fake news, in the retweeting domain. Starting from our 159 initial Twitter accounts and their geolocation data, we could associate every mention and retweet link to its account's geo coordinates, using a lookup table made by us[1]. Then we used *Cartopy* [8] in order to represent all the links inside an Europe map[2], exept for the self loops, that were too confusing to be represented.
In total, we created four maps: Figure 3.1 contains all the links taken from the mentions and retweet links, while the other three contain just a specific intra-class link, that is links where every pair of users belonged to the specific class. The last two maps (Figure 3.3 and Figure 3.4) are the emptiest ones, indeeed there were not much users linked to any other accounts (actually links were more, but a lot of them were self loops): we indeeed represented also the *lonely users* as a dot withount any link attached to it.

For the second geolocation analysis, we took a subset of about 10 tweets published by every `fnp` account in our list, and we retrieved then sorted the geolocation data from all the accounts who retweeted the former set (we gathered about 2k unique users). In this way we were able to categorize all the accounts based on their region of provenience. Actually, not all the accounts that retweeted the initial set had a significant geolocation value (sometimes it was simply not present, in other cases it was a troll or non sense value), so that we could not take them into account.
Finally, we represented of all the significant geolocation data we gathered (they were about 65% of all the items inside the legit geolocation subset; the rest were

[1] We used [7] to find the geo coordinates for each city.
[2] We used an Europe map, because a bunch of users were associated with a non spanish geolocation data.

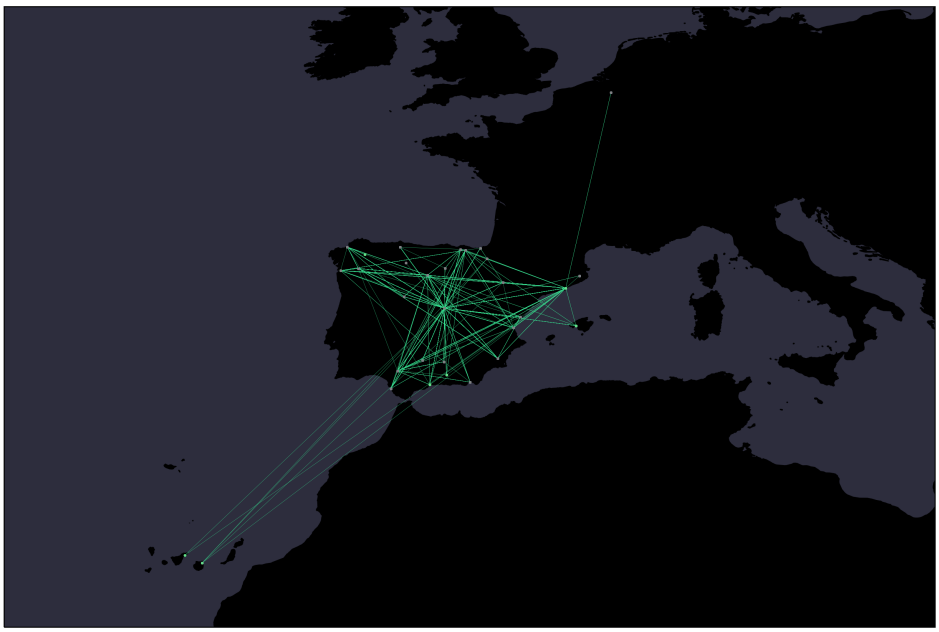


Figure 3.1: The figure shows all mentions and retweet links between nodes of our dataset.

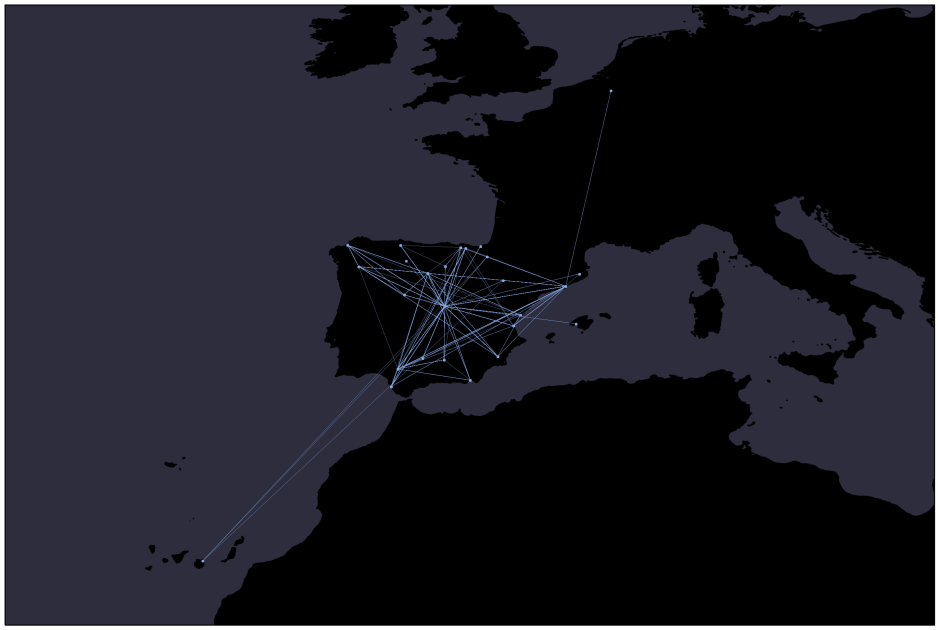


Figure 3.2: The figure shows mentions and retweet links between nodes belonging to the `pol` class.

Figure 3.3: The figure shows mentions and retweet links between nodes belonging to the `news` class. The dots are accounts which have not any intra-class link.

Figure 3.4: The figure shows mentions and retweet links between nodes belonging to the `fnp` class. The dots are accounts which have not any intra-class link.

simply outer-Spain locations). We splitted Spin into regions and we assigned a percentage to each one of them, based on the number of `fnp`-retweeting accounts that came from that specific region. The results are shown in Figure 3.5

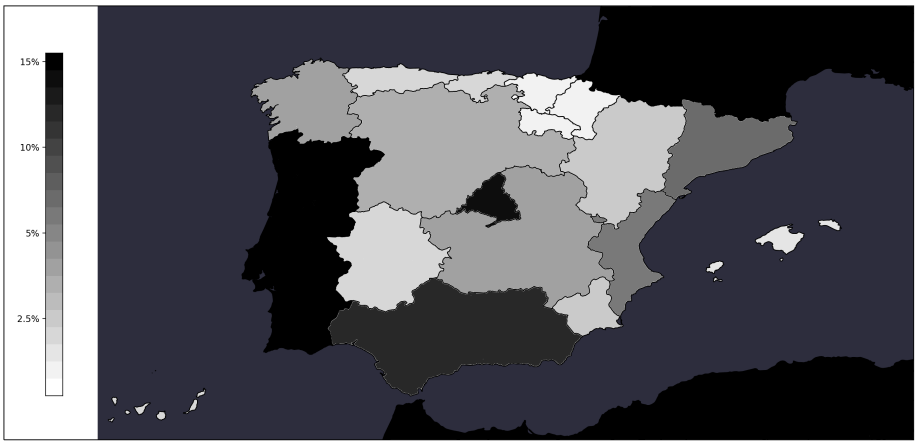


Figure 3.5: The figure shows the state of Spain divided into regions. Each region has a different shade of grey, based on the percentage of `fnp`-retweeting accounts. The darker is the color, the more `fnp`-retweeting accounts there are inside it.

3.1.1 Analysis Discussion

Figure 3.1 contains all the links taken from the mentions and retweet links. From this figure we can tell that the main political interactions come from the center, Madrid, also the capital, where we can find the government and the principal political institutions, and Its being spread around the rest of Spains peninsula, the Balearics and Canary Islands. We can also see a link to Bruselas.

In Figure 3.2 we can also see that the political accounts comes mainly from the center and that they have links all around Spain. We can deduce that Madrid is the main region for any political activity and that its position as the capital of Spain plays an important role as a news provider and with twitter accounts interactions for our analysis.

From this two figures we can analyses that the main starting point for news is Madrid which is linked to Sevilla and Bilbao, which means that theres has been some type of interaction between the nodes, we can also find nodes in Palma de Mallorca, the Canary Islands, the nord of Spain and the east coast that are not linked.

Compared to the map that shows the mentions to fake news providers we can tell that there is no explicit relation geographically with the spread of fake news. We can find four nodes, two of them are linked (Madrid and Vigo) while there isnt any relation with the other two (Murcia and La Corua), even though we can find nodes of news mentions in both regions.

Figure 3.5 shows the state of Spain divided in regions. Each region has a dier-

Figure 3.6

ent color, based on the percentage of twitter accounts that retweets fake news providers tweets. The darker the region, the more fake-news-retweeter accounts there are.

Comparing Figure 3.5 with Figure 3.6, we can tell that the regions with more fake news retweeters are Madrid and Andalucia. Knowing that the political parties that are more involved in fake news are PP and Vox, we can deduce that there is no relation between the amount of votes a party got in a region and the amount of fake news involved with a party since the winner party both in Madrid and Andalucia was PSOE.

3.2 Degree Composition Analysis

The *degree composition analysis* is a classification of our dataset users, based on the `pol`, `news` and `fnp` feature.
According to the explaination made in the previous chapter, we used a dataset

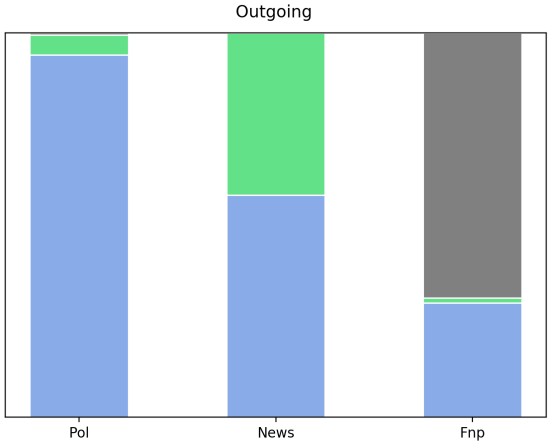


Figure 3.7: The figure shows the outgoing degree class composition of our dataset accounts. We sorted the mentioning and retweeting accounts in the three classes, one for each labeled bar. Every bar contains the class composition of mentioned and retweed users. Blue indicated the fraction of `pol` accounts, green are `news` and grey are `fnp`.

composed of 5531 links, that mapped each mentioning and retweeting account to its respective mentioned or retweeted one. Now, since every user is assigned to one and only one of these three former mentioned classes, we could enstablish a relation between classes of accounts, and study their behaviours in our dataset envroiment.
We splitted this analysis in two part: the *outgoing* and the *ingoing* degree composition. The first one is about the classification of mentioned and retweeted accounts: we took all the mentioning and retweeting accounts and for each one of them we recorded the class composition of users that are linked with them (these are the mentioned and retweeted accounts we wanted to analyse.). The results are stored in Figure 3.7.
The second analysis one is the very same thing, but the we analysed the class composition of mentioning and retweeting accounts. The results are in Figure 3.8.

3.2.1 Analysis Discussion

As far Figure 3.7, we can tell from the first and the second bar that the politician interact mostly through twitter with other politicians and news providers, while news providers also interact with the politicians and another news providers. In the last bar, we can see that Fake news providers interact mostly with other twitter accounts that provide fake news, they have some interaction with politicians, and to a lesser extent with news providers. Which lead us to the conclusion that each community interact between them, avoiding outgoing links with the fake

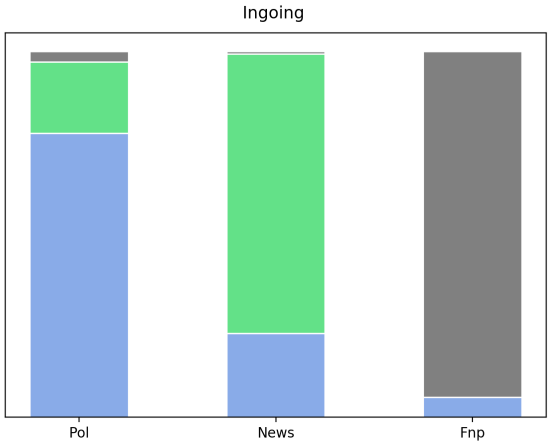


Figure 3.8: The figure shows the ingoing degree composition of our dataset accounts. We sorted the mentioned and retweeted accounts in the three classes, one for each labeled bar. Every bar contains the class composition of mentioning and retweeting users. Blue indicated the fraction of `pol` accounts, green are `news` and grey are `fnp`.

news providers.

As far Figure 3.8, we can see in the first bar that the politicians accounts receive nodes from another politicians and news providers accounts mainly, and in a lesser extent from fake news providers accounts. News providers also interact mostly with their same community and politicians, while fake news providers, as we can see in the third bar, receive nodes from another fake news providers, and politicians secondly.

3.3 Engagement Analysis

The *engagement analysis* shows a metric related with the number of retweets belonging to each tweet in our dataset: the higher is the expected value of retweets, the higher is the account's engagement.
We split the analysis in two parts. For the part one we used the complete tweets dataset, containing all the 33k tweets we retrieved from the initial list of 159 users, and the respective retweet count for each tweet. For each class of users, we created a graph very similar the the degree distribution one we used during lessions: in the x-axis we put all the unique retweet count we found in tweets belonging to each user from a certain class; in the y-axis we put the probability related to each retweet count value. Then we compared the `fnp` graph respectively with the `pol` and the `news` one, with an overlay. The results are plotted in Figure 3.9.

The second part of the analysis is a relative comparison, based on accounts that have a similar follower count. We divided all our 159 accounts in four

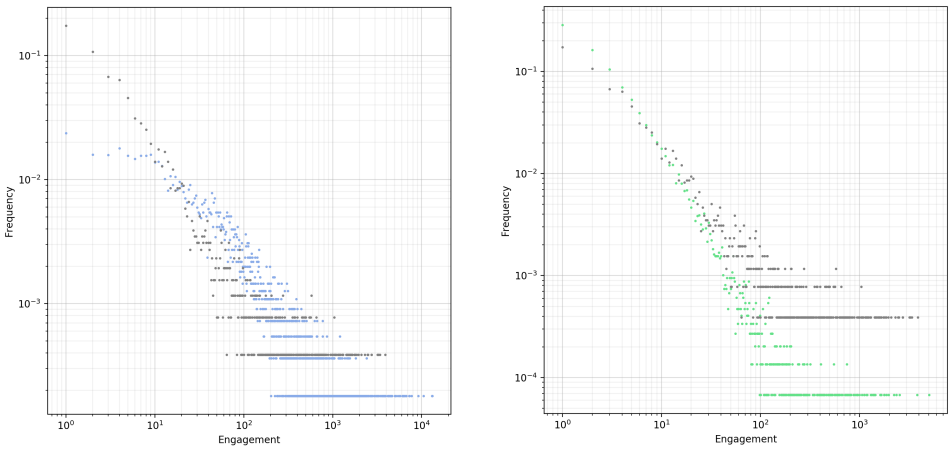


Figure 3.9: The figures show the absolute engagement logaritmic plots. The left figure shows a comparison between `fnp` tweets (in grey) and `pol` tweets (in blue). The right figure show a comparison between `news` tweets (in green) and `fnp` tweets (in grey).

groups, based on the order of magnitude of each account's follower count: in this way we could make a more precise engagement comparison between users from different classes. For each group we compute the average retweet counts, using all tweets which happened to be in that category, and we plotted that information in the upper y-axis of the graph. The lower part of the y-axis is just the accounts count for each group. The results are displayed in Figure 3.10.

3.3.1 Analysis Discussion

In the first graph we can see that fake news providers tweets may have a lot of RT counts but dont manage to get a lot of probability regarding being retweeted, which means that there is not a lot of engagement, even though the politicians accounts are less likely to be retweeted. The second graph shows that the news providers are more likely to be retweeted and compared to fake news providers, get more retweets, which shows a higher level of engagement.

3.4 Network Analyses

Given the initial nodes of the network, in the Modularity and Robustness analyses, an algorithm has been applied which removes the nodes that are not connected to any other node, i.e., the nodes whose sum of the in and out degree equals zero, and the nodes which are not connected to the giant component. In the so-called giant component for every couple of nodes theres a path connecting them.
Using this algorithm, our network, in the beginning constituted of 159 nodes,

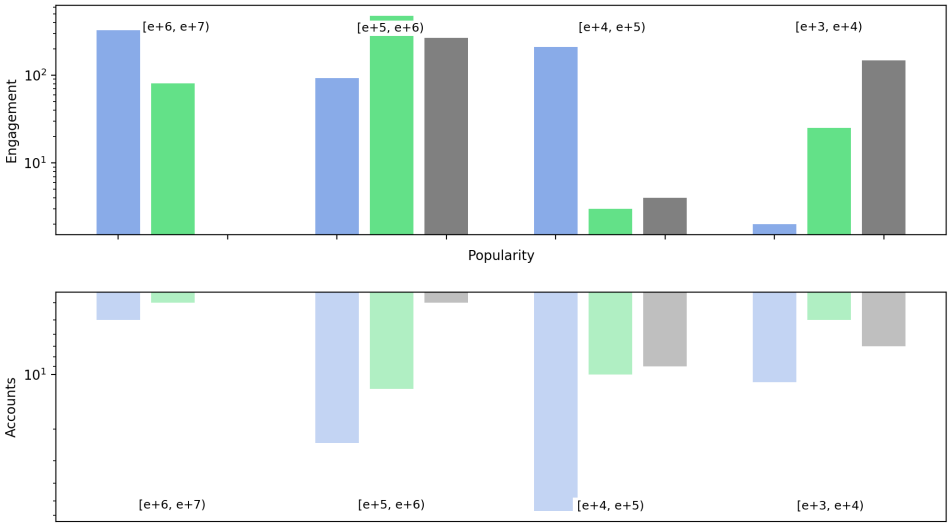


Figure 3.10: The figure shows the relative engagement. The color used are the same of the absolute engagement: `pol` average retweet count in blue, `news` in green and `fnp` in grey.

became composed of 93 nodes, in which there is no more the UP party. It is composed as follows:

- 66 nodes of politicians profiles: 15 for PSOE, 16 for PP, 15 for VOX, 20 for CS
- 21 nodes of official news profiles
- 6 nodes of fake news providers (fnp)

3.5 Modularity Analysis

Firsty, we want to cluster our network, to see if there are evident relations between the politicians and the fnp. To cluster our network made of 93 nodes a suited algorithm for directed networks described precisely in [9] is used. A modularity algorithm studies how different our network is from a random network, where the nodes and their in and out degrees are the same, but with the links placed randomly among them (Molloy-Reed model). The difference between the edges within groups into our network and the edges under random rewiring, normalized to the number of total edges of the network gives the so-called modularity value Q. Taken from [9], the modularity is thus obtained in this way:

$$= \frac{1}{m} \sum_{i,j} \left[A_{ij} - \frac{k_i^{out} \cdot k_j^{in}}{m} \right] \delta(c_i, c_j)$$

$$= \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i^{out} \cdot k_j^{in}}{m} \right] (s_i s_j + 1)$$

$$= \frac{1}{2m} \sum_{i,j} B_{ij} s_i s_j$$

$$= \frac{1}{2m} s^T B s$$

where:

$$B_{ij} = A_{ij} - \frac{k_i^{out} \cdot k_j^{in}}{m}$$

and $s_i = 1$ if the node i belongs to the 1st group, $s_i = -1$ if the node i belongs to second group. Working with directed networks, the problem using this formula is that B is not symmetric. To work with a symmetric matrix the following formula for calculating Q is then applied:

$$Q_d = \frac{1}{4m} s^T (B + B^T) s$$

If the modularity Q is greater than 0 it means that the network can be clustered and the 2 communities identified by the vector s clusters well the network. This method can be used if we know the clusters and we want to compute the associated value of the modularity Q. But what if we dont know the communities? We search for the vector s that maximizes Q. To do so we compute the eigenvalues of the symmetric matrix obtained from the sum of B and B transposed, we look for the larger eigenvalue and we take the eigenvector associated to it. The entries of this vector will be surely real, since we are working with a symmetric matrix, so we look for the sign of the values: if the value of the entry j is positive, then node j belongs to the 1st component, otherwise, if the value is negative, the node j belongs to the second component.

Knowing the theory, our approach was to divide the network into 2 subnetworks, which we can call $G1$ and $G2$ and compute the relative modularity Q. If this value is at least 0.3 it means this is a good clustering and therefore $G1$ and $G2$ can be kept as 2 subnetworks of the initial network. Then we iterate the process on $G1$ and $G2$, to obtain altogether 4 subnetworks. Compute the 2 modularity values associated with these clusterings. If these are greater than the value obtained in the first step the clusters can be kept. The process continues until the value of Q keeps increasing, otherwise declare the subnetwork indivisible and keep it merged.

This algorithm was applied in overall network, but we also tried to apply the modularity on the network made only of 66 nodes of politicians. This time we knew the groups in the network, so we knew the vector s. Therefore, we computed directly the relative modularity to see if the real data matches our expectation.

Figure 3.11: Clustering on the network made by the modularity algorithm described in [9].

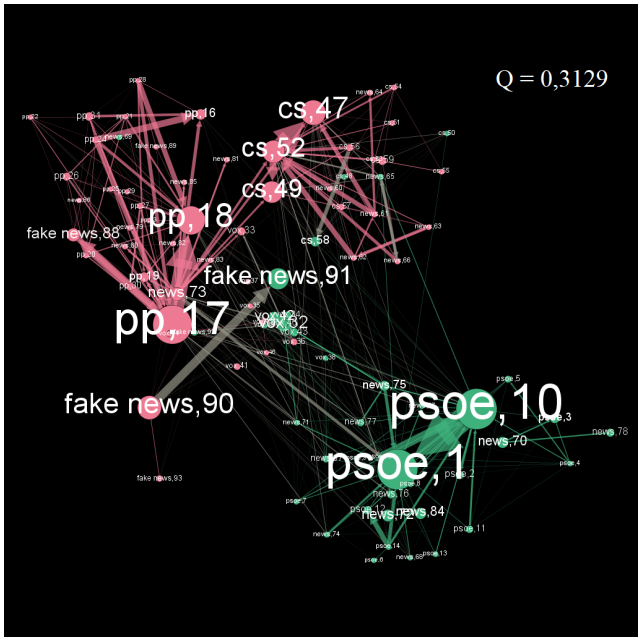


Figure 3.12: Clustering on the network made by the modularity algorithm implemented into *Gephi*.

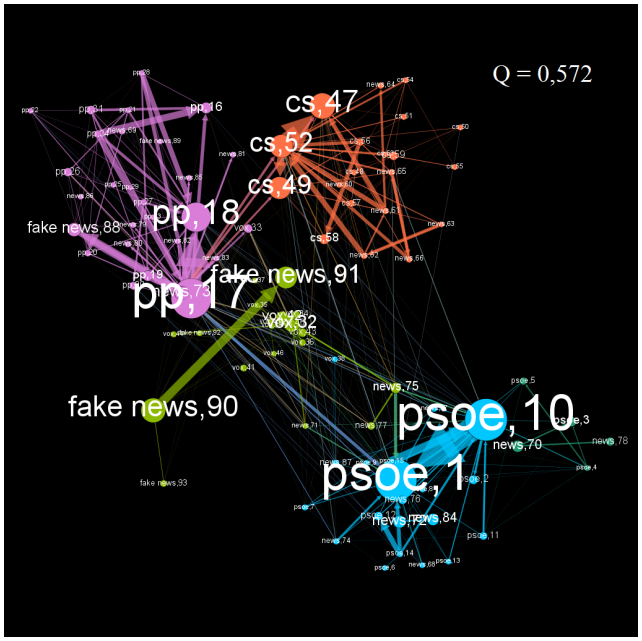


Figure 3.13: Clustering on the network made of only politicians profiles.

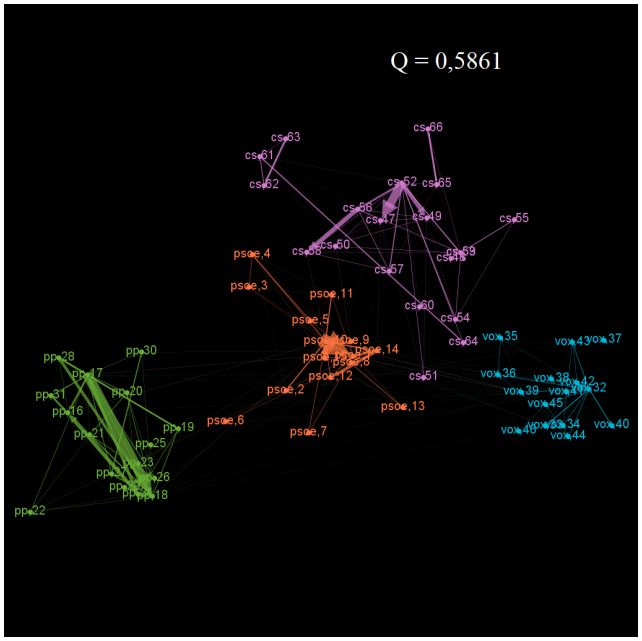
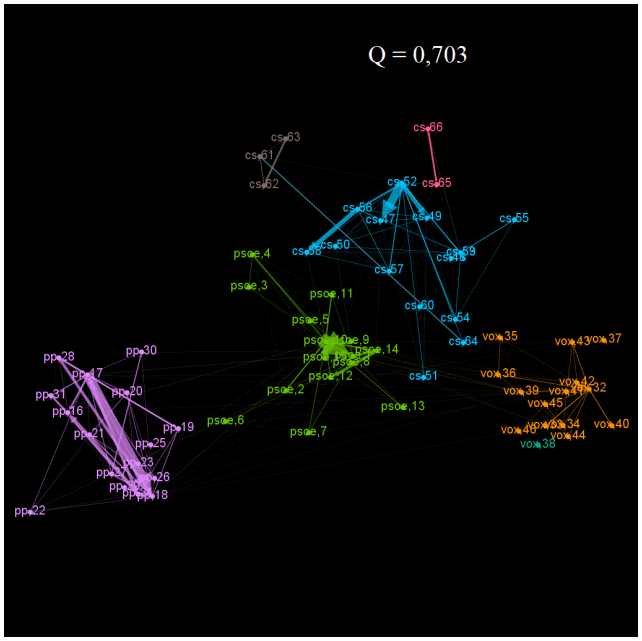


Figure 3.14: Clustering made on the network of only politicians parties by the *Gephi* algorithm.



3.5.1 Analysis Results

The images represent the clusters of the different networks. The nodes with the same color are in the same group. In Figure 3.11 and Figure 3.12 it is presented the network with the 93 nodes clustered and the relative Q obtained.
The clustering in Figure 3.11 is a result of the algorithm that was described above. The 2 subnetworks were further divided but the Q obtained in this way was lower than the initial one, so the initial network was clustered only into 2 groups. was obtained through the algorithm used by *Gephi*, which is based on the idea presented in Figure 3.12. It is evident that the *Gephi* algorithm is way more effective since it finds more clusters and the associated value of modularity obtained is higher than the one written in Figure 3.11.
Figure 3.13 and Figure 3.14 show the clustering made on the network with only the politician profiles. In Figure 3.13 it is used the algorithm described in the last few rows of the previous section, whereas Figure 3.14 shows the clustering done by the *Gephi* algorithm.
Here the results are better, for we know the clusters and we just compute modularity. Indeed, we see that the value of modularity in Figure 3.13 is bigger than the one in Figure 3.11 and the value in Figure 3.14 is bigger than the one in Figure 3.12. In Figure 3.14 theres still a value of Q greater than the in Figure 3.13, and this is because Gephis algorithm found 3 more clusters: indeed, we can see that in Figure 3.14 there are overall 7 seven different colours. If we look even more carefully we see that 3 different colours are used to cluster 3 very small components. Two of these clusters are subnetworks of the Cs cluster, and the other is made of only one node: *vox,38*. What is happening is that just considering the network made of only the nodes of the politicians profiles means cutting the links to the nodes of news and fnp. In fact, looking to the adjacency matrix, we have seen that this *vox,38* was linked in both directions to an fnp so, cutting all the links to the news and fnp, this node became isolated.

3.5.2 Analysis Discussion

- Comm1. We can see clearly two main communities. Same colour means they are in the same community, so we can see PP and Cs are in the same one, and there are actually links between them. PP is linked to some little fnp. In the other community theres PSOE with a lot of links with news nodes and with Vox. These two political parties are from opposite political sides, since PSOE is a left-wing party and Vox from the extreme right-wing.

- Comm2. In this image the communities are divided by political parties, news and fake news providers. We can differentiate three big communities: PP, Cs and PSOE. We see PP and Vox are the only two communities linked to fake news providers, and that the two biggest nodes of fake news provider are in the same community of Vox. This is not a surprising result and makes sense, since PP is a right-wing political party and Vox an extreme right-wing party. In fact, Vox was born because some years

ago some politicians of PP got separated from this party to create their own one. Apart from this, Cs and PSOE are linked to news nodes, and both of them have links to PP.

- Comm 3 and 4. This images illustrate communities without fake news providers, and it shows that every community is isolated and there is not interaction between them.

3.6 Robustness Analysis

The idea behind this analysis is based on our intention to explicit the correlation of the elections to the fake news spreading. Firstly, we thought that to see if there is a correlation between the political parties and the fnp we can remove the nodes of the political parties from the network and watch how the network changes. Moreover, based on our intentions, we want also to correlate this network transformation to the elections. Therefore, what we firstly did was to gather all the data concerning the elections results of 10th November 2019. We took the votes and we kept them split on the 19 regions of Spain. We are studying the impact of the fake news on the 5 main political parties in Spain, but we found that there are other parties, and some of them gained a good number of votes in some regions.
To keep the analysis simple, we then assumed that there were only these 5 parties, taking only these votes and assuming that the people who voted did so only for one of these parties. We restricted our range of parties considered, but doing so we normalized the number of votes considering the total votes of every region as the sum of the votes gained by the 5 parties. We got the data of the geolocalization and the number of followers of the profiles. The number of the followers of a profile can be a good measure to weight the importance of a profile, but we can be more precise using also the *Pagerank algorithm* [10].
The Pagerank simulates a random walk through the nodes, as if a user is surfing on the Twitter pages. Assume that at time t a random surfer will be in a node U and from that one he will go to one node U points to. Where he will be at time $t + 1$ depends on how many links the node U has to the different nodes it points to. We can then describe the probability of getting to a neighbour T of U as the number of links from U to T normalized to the out degree of U. To describe this process for the whole network we need to define M as the adjancency matrix A, but with the columns normalized to their sum, i.e., the out degree of every node. We then define the vector p, composed of 93 values, which is the number of our nodes. Our equation will then be the following:

$$p_{t+1} = c \cdot M \cdot p_t$$

Where $c < 1$ and its called the damping factor.
To get a more effective rank for our nodes we also add a so-called teleporting factor, which includes the possibility for the random surfer to get randomly to

a page which is not pointed by the ones it is at time t:

$$p_{t+1} = c \cdot M \cdot p_t + (1 - c) \cdot q$$

It can be proved that iteratively applying this formula, it converges to the solution of the equation. The output of the algorithm will be thus a vector consisting of 93 entries. It is a stochastic vector, that is, the entries sum up to 1. We can see the entry i of the vector as the probability that a user ends in the page i, so in the Twitter profile labeled with i.

Now that we had the number of the followers, the geolocalization of the profiles, the rank of the profiles, the votes for the different parties in every region, we could put them together to fulfill our intentions. Nevertheless, how could we do that? We imagined that the contents of a politicians profile affects some people to vote for the party the profile support. In other words, we can think that every politicians profile carries some votes, and by removing it, we remove those votes. Therefore, we thought of an algorithm that simulates a sort of attack towards a single party.
This algorithm would implement also a function that, given the parameters above stated, could compute a number of votes. To include the dependence on the fnp, the output of the function will then depend also on the number of links the profile has to the fnp, but we will discuss better the function later. The number of votes given as output from the function would then be removed from the region where the profile is located, and the votes removed would be equally splitted into the other parties. To better see how every party behaves when attacked, we focus on one party at a time. From this idea we built an algorithm that works in the following way.

Focus and fix one party and for that one do the following:

1. find the node of the party which has the highest rank;
2. gather its information which are: its geolocalization, its number of followers, its rank, its number of links pointing to the fnp;
3. calculate the number of votes through the function introduced above;
4. if the geolocalization data can be used, that is, if it is not *Spain* or *Brussels*, but it refers to a region of Spain, remove the number of votes computed from the region where the profile is located;
5. update the network and recalculate the new rank without the removed node;
6. repeat until no nodes of the party remain.

In this way we are assuming that the profile we are removing is active and carries votes only from his region, which is indeed a strong assumption, since obviously a Twitter profile can be seen throughout Spain and all the world.

3.6.1 Building the num-votes function

We have already stated what are the characteristics that we want to have: it has to have a dependence on the number of followers, the rank and the number

of links the profile has to the fnp. After a few steps we thought about this formula:

$$num_{votes} = pop_{twitter} \cdot \frac{num_{followers}}{followers_{parties}} \cdot rank \cdot (LinkstoFNPs \cdot w_{FNP} + 1)$$

The interpretation of the formula follows. We recall that the rank of a profile can be seen as a probability for a user to end up in that profile web page. The constant $pop_{twitter}$ is the twitter population that follows the elections of the 10th november in Spain. Its value will be discussed later. The $num_{followers}$ is one of the values given in input; $followers_{parties}$ is the sum of all the followers of all the political parties considered in our analysis, and it is fixed too; the rank is given as input as well as the value $LinkstoFNPs$; w_{FNP} depends on how much we want to weigh the fnp inside the analysis.
The value:

$$\frac{num_{followers}}{followers_{parties}}$$

can be seen as the probability that a person is a follower of the profile we are considering. The rank itself is a probability of ending up to that page, and we can see it as a probability that a follower will vote for that person (for that party) he is following.

As an example, let us then fix a profile P: the formula is thus a probabilistic prediction of how many people following P will vote for P (so for the party P belongs to) weighted on how many links P has to the fnp. If we fix $w_{FNP} = 0$, we do not consider the action of the fnp, if we put $w_{FNP} = 1$ and if P has a link to a fnp, the votes will be doubled with respect to the number of votes without the action of the fnp.
The constant $pop_{twitter}$ has been decided taking into account the people that actually voted in the elections of November. We considered a value we can call $pop_{present}$ as the sum of the people that follows the politicians parties, which is the value $followers_{parties}$ showed above, and half of the people following the journalists (fake or non-fake) profiles. This is because we consider that some persons that follow a journalist profile follow also a politician. Since we will use this same function for the analysis on the evolving network, we also considered the same sum pop_{future} but for the evolving network seen in the next section. Finally the $pop_{twitter}$ is given as follows:

$$pop_{twitter} = \frac{sum_{present} + sum_{future}}{2} =$$

$$\frac{(follower_{parties,present} + \frac{follower_{news,present}}{2}) + (follower_{parties,future} + \frac{follower_{news,future}}{2})}{2}$$

We obtain a value of $pop_{twitter}$ which is $pop_{twitter} = 20091364$.
We can get from internet [11] that more than 24.3 million of people voted in the elections. We are assuming in this way that among these 24.3 million the number of people corresponding to $pop_{twitter}$, which is the 82,5%, had twitter, followed one politician among the one were considering, followed maybe one news profile

(fake or non-fake), and voted for one of the 5 parties were considering.
We have indeed made a lot of assumptions to simplify our analysis. Here they are summed up:

- there are only the 5 main parties in the elections held on November 2019 in Spain;
- a politicians profile is active only in the region where the twitter profile is localized;
- a politicians profile brings a number of votes from the region it is active to the party it belongs to;
- every time we remove a node we assume it never existed, so we recalculate the rank on all the network minus that node;
- the probability rank of a profile, calculated through the Pagerank algorithm, is the probability of a person to vote for that profile (so for the party the profile belongs to);
- the number of the followers of a profile is not one to one correspondence to the people following that profile. This number is used to get the probability that a person follows this profile. It can be seen as the number of chances to win among the number of possible chances;
- $num_{twitter}$ people have twitter, follows a politicians profile of one of the 5 parties were considering and votes for one of the 5 parties;
- a profile whose geolocalization is not valid does not bring any vote to the party.

3.6.2 Analysis Results

The results of the 4 attacks are shown below in the figures. 3 different weights have been used for the fnp to better see how much they have influenced: the colors blue, red and yellow have been used to show the transformation on the network with the fnp weight of respectively 0.5, 1 and 1.5. In Figure 3.15, which shows the attack to the PSOE, we see that theres only one yellow line: thats because PSOE has not any link to the fnp, so the transformation on the votes is the same whether we consider or not the action of the fnp, and we weight more or less their influence. The line starts immediately with a high negative slope, and then gets almost on the same level of number of votes. The high initial slope is because we are removing the 2 main politician profiles in the beginning. Figure 3.16 shows the attack to PP, which is particular because after just one elimination, the votes, which started from around 5 million goes a lot under the value 0. Inspecting the adjacency matrix we discovered that the first node eliminated, which corresponds to `@populares`, has 60 links to a fnp! This obviuosly affects num-votes function which gives a value of votes to be removed very high. Indeed, the population of twitter considered in this way is 61 times the initial

Figure 3.15: Attack to the PSOE party.

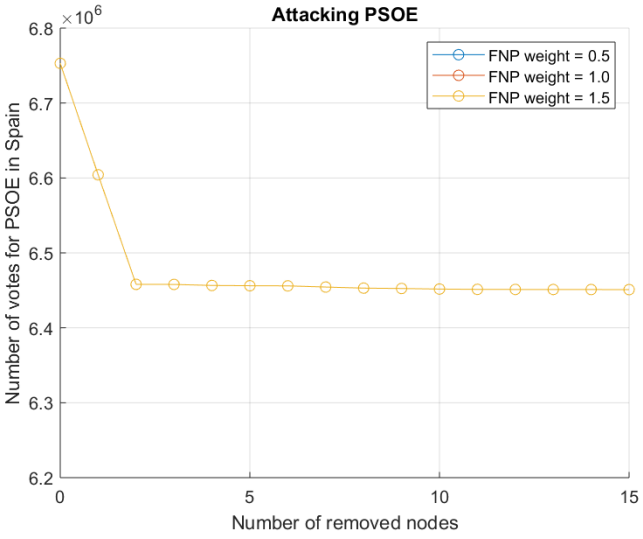


Figure 3.16: Attack to the PP party

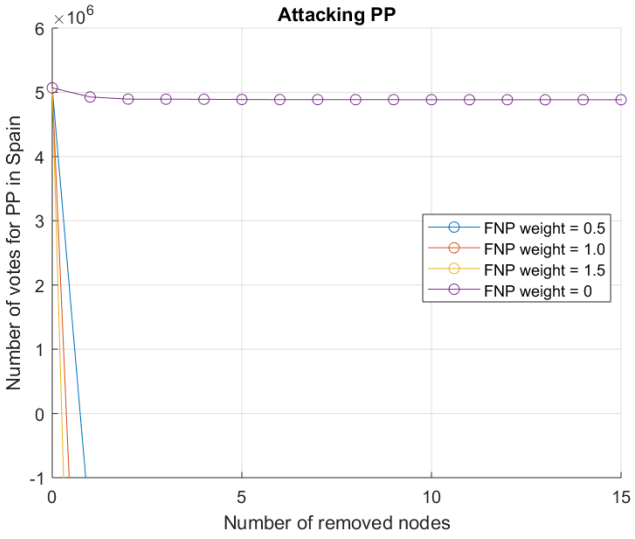


Figure 3.17: Attack to the CS party

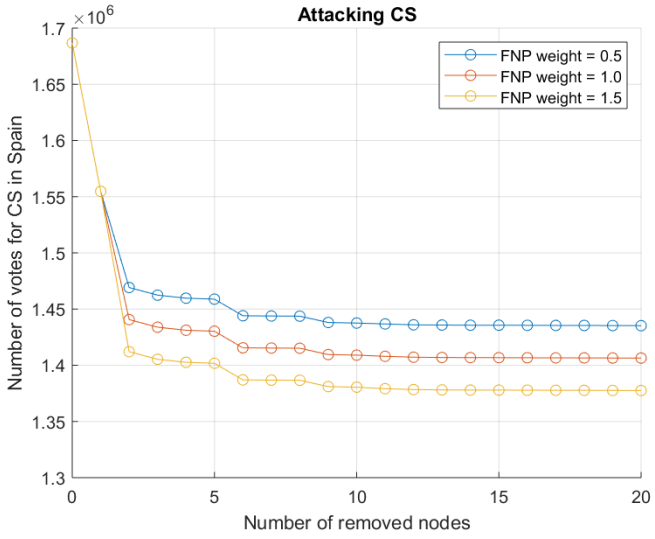
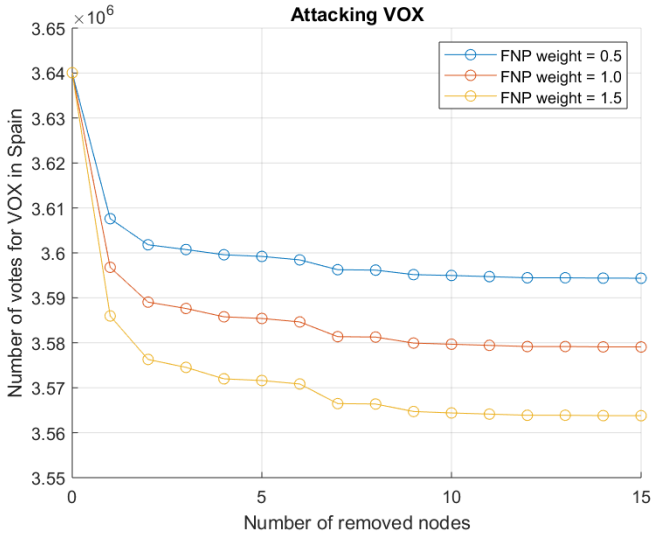


Figure 3.18: Attack to the VOX party



one! It is as if, for this node, together with the $pop_{twitter}$, we are adding other 60 $pop_{twitter}$ 1.2 billion of persons to the analysis.
Concerning Figure 3.17 and Figure 3.18 we see that at most, removing all nodes, the least variations in number of votes are respectively about 200000 and 40000, and at most about 300000 and 80000.

3.6.3 Analysis Discussion

We focused on the analysis of the results of the line FNP weight = 1,5

- PSOE. When the first two nodes are removed, the number of PSOEs votes is decreased by 294.974. This number represents a 4,37% of their votes and 1,25% of Spanish votes.
- PP. The case of PP is the most particular one. Whatever the FNP weight is, after removing the first node goes high negative values. This is because this node corresponds to the party main account on Twitter (@populares),
- Cs. After removing the two nodes theres a decrease of 274.667 votes, representing a 16,28% of the total votes this political party got in the elections; and 1,17% of all the Spanish votes.
- Vox. In the case of this political party, removing the first two nodes doesnt have that much impact, since they represent 1,75% of the votes of Vox and 0,27% of the Spanish votes.

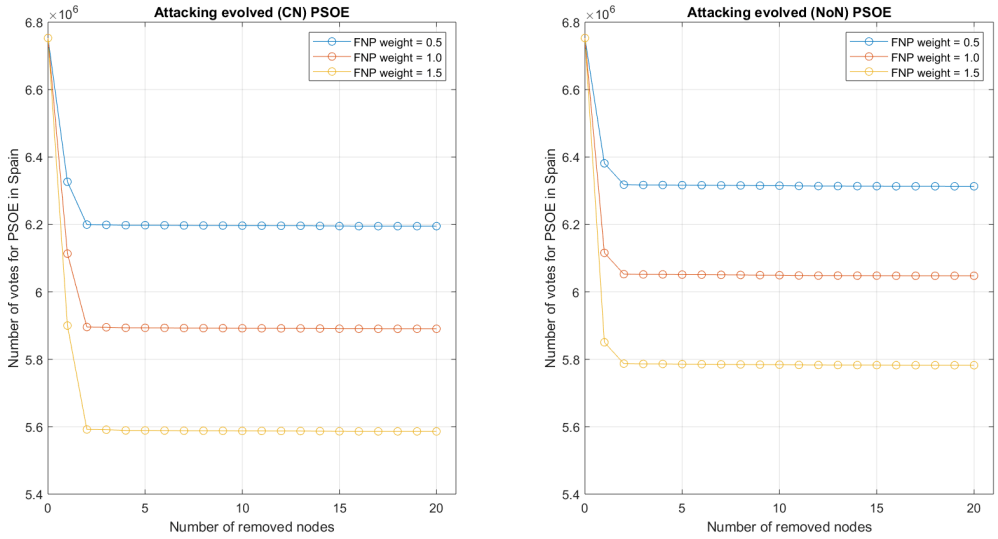
After getting all these results, we can see in the cases of PSOE, Cs and Vox that after removing the first nodes, the lines remains stable.

3.7 Evolving network

What would have happened if more links had been created among our nodes? What if there was more time and the network could evolve? Would the fnp had more impact on the elections? To answer these questions we took the initial network, the one made of 159 nodes, and we applied 2 different simple algorithms to link the nodes among them. The 2 different algorithms correspond to 2 criteria used to create the new links. The first one is based on the fact that if 2 nodes share frequently the same content, they have a good probability that in the future they will link together. The second criterion derives from the idea that if a lot of profiles that a profile U points to, point to a profile T, it is likely that U will link to T in the future. We can sum up the two criteria as the one based on the common neighbours (CN), and the one based on the neighbours of the neighbours (NoN).
We then computed 2 different matrixes, S_{CN} and S_{NN}, built from A, the starting adjacency matrix:

$$S_{CN} = A^T \cdot A \qquad\qquad S_{NN} = A \cdot A$$

Figure 3.19: Attacks to the evolved networks of PSOE. On the left, the graph representing the attack to the network evolved through the CN criterion, on the right the attack to the network evolved with the NoN criterion.



S_{CN} will have have in the entry (i, j) the number of common neighbours between the nodes i and j. S_{NN} will have in the entry (i, j) the number of the nodes pointed by j, which points to i. In both the matrixes the indexes corresponding to the 2 maxima are found for every column and on every row. Using these indexes 2 incoming links a 2 outgoing links are added for every node. In this way we are building overall 4 links for every node, the 2 incoming and 2 outgoing most likely to happen in the future. After this step the same analysis of robustness is performed on the resulting matrixes.

3.7.1 Analysis Results

The results are the graphs which show the attacks on the evolved networks. On every figure the left graph corresponds to the attack to the network evolved through the CN criterion, whereas the right one corresponds to the attack to the network evolved through the NoN criterion.
The reader should not look at these graphs singularly, in fact, these graphs are useful when compared to the versions seen in the previous section.
Before we compare the figures, its important to grasp some considerations that link this study to the previous one: firstly, one can now understand why the $pop_{twitter}$ constant has been kept the same for the two studies, and why it depends on both the analyses on the reduced network and on the evolved ones; it is because we can better compare the results; moreover, even though the $pop_{twitter}$ is constant for the two studies, an inevitable variation has to be counted here.

Figure 3.20: Attacks to the evolved networks of PP. On the left, the graph representing the attack to the network evolved through the CN criterion, on the right the attack to the network evolved with the NoN criterion.

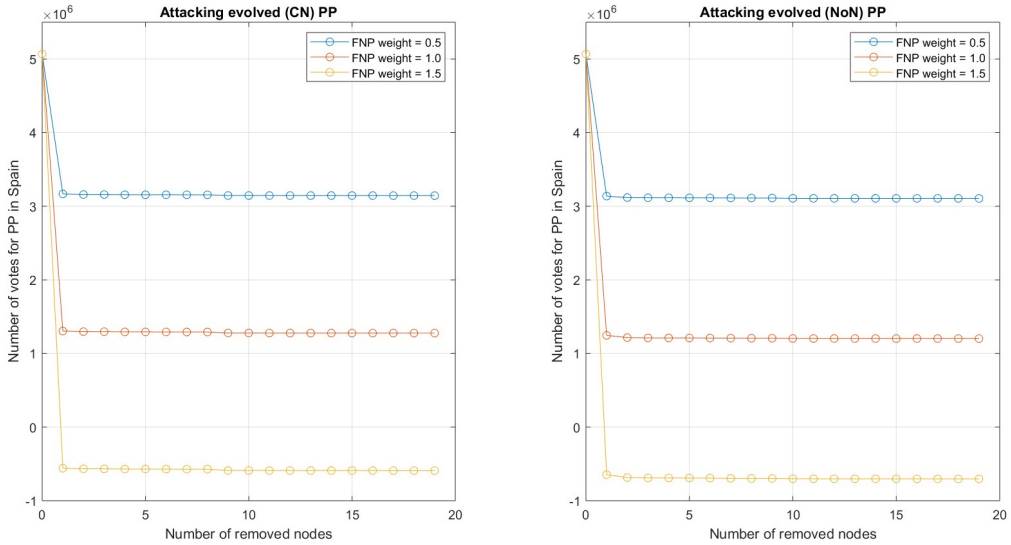


Figure 3.21: Attacks to the evolved networks of CS. On the left, the graph representing the attack to the network evolved through the CN criterion, on the right the attack to the network evolved with the NoN criterion.

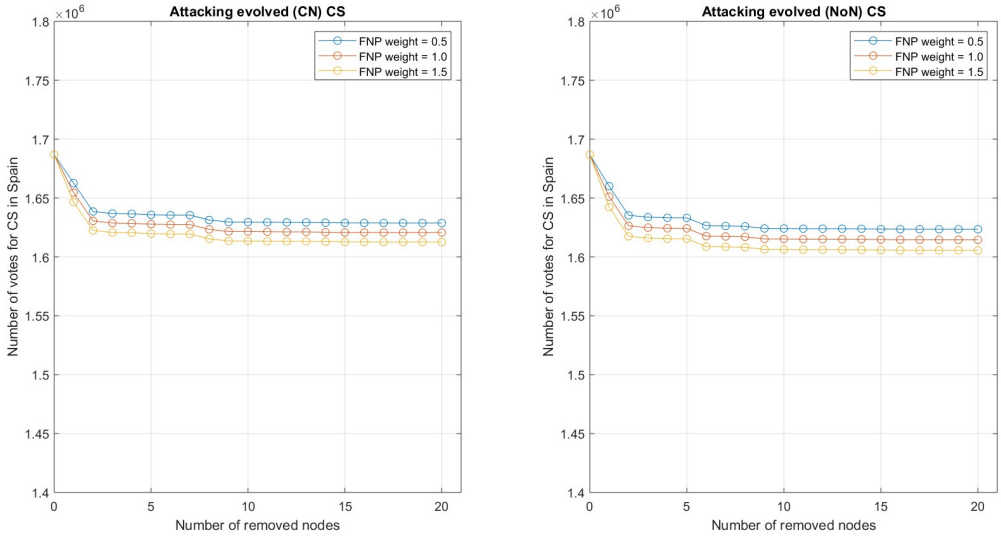


Figure 3.22: Attacks to the evolved networks of UP. On the left, the graph representing the attack to the network evolved through the CN criterion, on the right the attack to the network evolved with the NoN criterion.

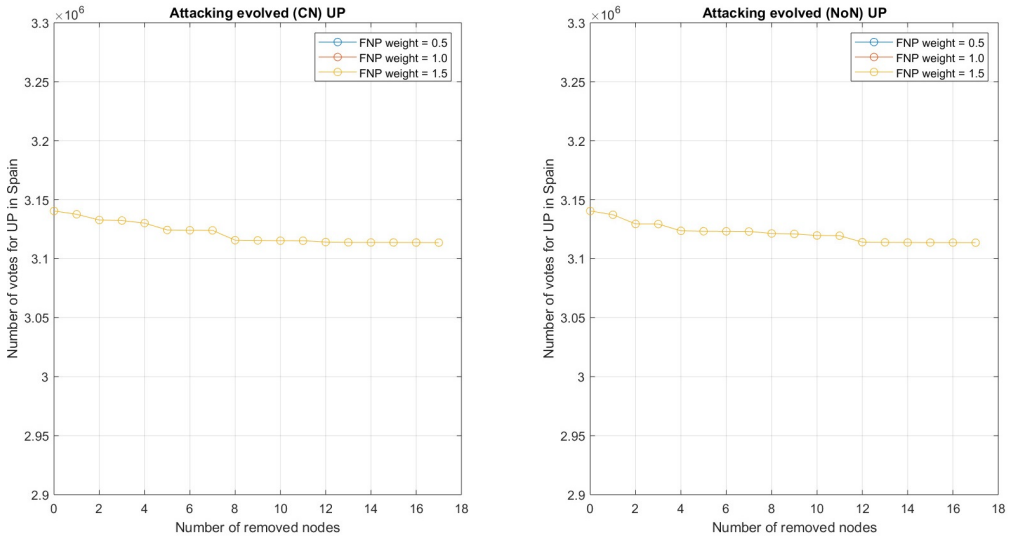
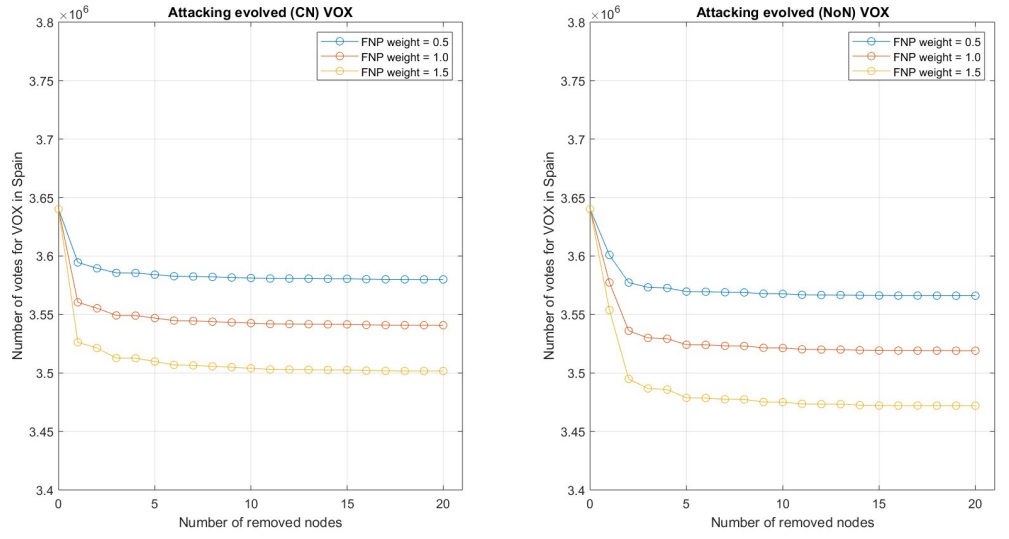


Figure 3.23: Attacks to the evolved networks of VOX. On the left, the graph representing the attack to the network evolved through the CN criterion, on the right the attack to the network evolved with the NoN criterion.



This variation concerns the probability of one person to follow a politician: indeed, here the politicians are more in number than in the previous study; this makes the constant $followers_{parties}$ increases. Naturally, this implies that the probability for every party profile gets lower, since as the number of possible chances (the overall number of followers of the politicians) increases, the number of the chances to win (the number of followers of a specific politician) are the same. It is true we are adding more political profiles, but this increase is not enough to overcome the lower probability. So, in general, for every party, the decreasing of votes is less than before.
Another thing must be counted here: keeping the nodes of the initial network means also to keep the initial links; this implies that a possible increase in number of links to the fnp from a party does not mean that these links have been added through the algorithm. The new links to the fnp could have been active in the initial network so they appear now as we look at the whole network. Let us now compare the graphs.

Firstly, comparing Figure 3.19 with Figure 3.15, we immediately discover that the evolved PSOE has some links to the fnp. We can see this because in Figure 3.19 three lines appears, instead of the single line seen in the robustness analysis. We also recognize that here the attack brings the votes lower than in the robustness analysis. This is because the increase in number of fnp can overcome the decrease of probability.
Moreover, the decrease to the CN network is bigger than in the NoN network. Inspecting the adjacency matrix we can see that in the reduced matrix, the one of the previous section, the number of links from the PSOE to the fnp is obviously 0, as also seen in the Figure 3.15, whereas the number of links to the fnp from the evolved CN PSOE is 8 and the number of links from the evolved NoN PSOE is 4.
Comparing Figure 3.20 with Figure 3.16 and Figure 3.21 with Figure 3.17 we see that there is no big increase in number of links to the fnp as the votes has a smaller variation compared to the one in the reduced network.
Concerning the UP party, the one that was removed by the algorithm in the robustness analysis, what we can say here looking at the Figure 3.22 is that even applying 2 different criteria to add some links to and from this party, it has not links to fnp. This implies that considering the network not evolved, so the one that describes the real elections of 10th November 2019, we can state that this party had not links to fnp.
Finally, concerning VOX party we see that Figure 3.23 and Figure 3.18 has one similarity with the graphs of PSOE: the decrease of votes gets higher in the evolved network, in VOX party the biggest variation seen in the NoN evolved network, whereas in PSOE is in the CN evolved ones. Inspecting the adjacency matrix here as well, we can see that in the reduced network the number of links from VOX to the fnp is 13, whereas in the CN and in the NoN they are respectively 32 and 31.
In this case an interesting thing happens. Even though the evolved CN network presents more links from VOX to the fnp that in the NoN ones, the NoN evolved VOX causes a bigger decrease in votes: inspecting the adjacency matrix,

and precisely the links from the single nodes of VOX to the fnp, we see that in the NoN evolved VOX there is more concentration of the outgoing links to fnp in the profiles that have a bigger number of followers, whereas in the CN evolved ones the links are more distributed among also the VOX profiles that have less followers. Recalling our function num-votes this implies that when these numbers are put as input the number of votes computed is higher in the NoN case. From this point we also see that the decreasing in votes depends also on the distribution of the number of links to the fnp among the profiles.

3.7.2 Analysis Discussion

In case of having more time, fake news would have had a different impact on the elections results of every party.

- PSOE. In both cases (CN and NoN), results are really different, even for the lowest FNP weight of 0,5. Lets analyse that. For CN with FNP weight = 0,5: when we attack the first node it goes down from 6.752.983 to nearly 6.300.000 votes. That means a decrease of 450.000 votes, representing a 6,66% of PSOEs. Attacking the second node falls to 6.200.000, and after that remains stable. This happens to all the FNP weights, in both CN and NoN. Lets analyse the FNP weight = 1,5. After attacking the first node, votes fall down to 5.900.000, which means around 850.000 votes, representing a 12,59% of the votes this party got.

- PP. This is also a particular case if its compared with the robustness analysis. In this evolving network, after removing the first node votes decrease to numbers of around 3.100.000, 2.250.000 and (-500.000), in the respective order of FNP weights = 05, 1 and 1,5. From the second node, votes remain stable in that number.

- In the case of the other 2 political parties (Cs and Vox), attacking the nodes doesnt vary much from the study already done in robustness.

Having all these information we can say that in the hypothetical case that elections would have been later than the actual date, fake news wouldnt have had more impact on Cs and Vox; but would have changed a lot for PP and PSOE.

Chapter 4

Hypothesis and Conclusion

4.1 Hypothesis

When we first started this project, we had the following hypothesis: Vox will be the political party more related to the fake news, in the sense that they would be the ones spreading more fake news. But in contrast to our hypothesis, the results showed that PP is actually the one directly related to fake news providers. In fact, the 2 biggest nodes of fake news providers are: @TeoGarciaEgea, the General Secretary of the PP; and @PP-Villaverde, the twitter account of a district of the city of Madrid.

We thought Vox would be the political party closest to the fake news because it is a relatively new political party, from the extreme right-wing. From the very beginning, their speech was full of hate and absurd things. Everyone classified them as crazy. What nobody thought is that in the first elections of April 2019, they got high representation in the Congress, and they even doubled that representation on 10N.

4.2 Conclusion

During the process of making the project, we could see how easily it is to spread a fake new via twitter. A fake news provider could be a trustworthy news source, an anonymous account or even a politicians account. Even so, unlike the United States, in Spain the fake news hasnt proliferated that much. From our main political parties (PP, Ciudadanos, PSOE, PODEMOS and Vox) we concluded that only the right winged parties (PP and VOX) were linked to fake news providers and fake news as such, even though we could see links between PP and Ciudadanos, while Ciudadanos and PSOE are linked to news providers.

Also, fake news didnt affect the 2019 second elections as much as they could have. As we analyzed earlier, the elections result turned out positive if we compare it to the spreading of fake news around Spain. Which lead us to the conclusion that fact - checkers plays an important role today in order to maintain

population informed to avoid the influence of fake news in peoples political view. Even though Spain had to go through two elections in less than a year, we can tell that the results were fair.

Bibliography

[1] Wikipedia contributors, "November 2019 spanish general election — Wikipedia, the free encyclopedia," 2020. [Online; accessed 16-February-2020].

[2] Wikipedia contributors, "List of newspapers in spain — Wikipedia, the free encyclopedia," 2019. [Online; accessed 16-February-2020].

[3] M. Cinelli, S. Cresci, A. Galeazzi, W. Quattrociocchi, and M. Tesconi, "The limited reach of fake news on twitter during 2019 european elections," *arXiv preprint arXiv:1911.12039*, 2019.

[4] Media Bias Fact Check, LLC, "mediabiasfactcheck," 2019. [Online; accessed 16-February-2020].

[5] Twitter, Inc., "Twitter api docs," 2019. [Online; accessed 16-February-2020].

[6] Joshua Roesslein, "Tweepy," 2019. [Online; accessed 16-February-2020].

[7] LatLong, "Latlong," 2019. [Online; accessed 16-February-2020].

[8] SciTools, "Cartopy," 2018. [Online; accessed 16-February-2020].

[9] F. D. Malliaros and M. Vazirgiannis, "Clustering and community detection in directed networks: A survey," *Physics Reports*, vol. 533, no. 4, pp. 95–142, 2013.

[10] L. Page, S. Brin, R. Motwani, and T. Winograd, "The pagerank citation ranking: Bringing order to the web.," tech. rep., Stanford InfoLab, 1999.

[11] lavanguardia, "Elecciones generales noviembre 2019," 2019. [Online; accessed 15-February-2020].