# Social Network Analysis of Twitter Bots

Davide Cremonini, Artificial Intelligence, 0001137778
Alessia Crimaldi, Artificial Intelligence, 0001145505
Fabio Giordana, Artificial Intelligence, 0001145924
Gabriele Nanni, Artificial Intelligence, 0001146107

## 1 Introduction

In the current historical moment, the widespread diffusion of AI systems designed to simulate human behaviour is increasingly evident. This project aims to evaluate the effectiveness of these systems in the context of bot detection on Twitter, seeking to identify behavioural patterns that differentiate automated users (aka *bots*) from human ones. To this end, Social Network Analysis techniques were adopted.

## 2 Problem and Motivation

Nowadays it is clear the impact of social medias in people's everyday life. These applications are not only a way to interact with friends or observe celebrities, but they are becoming one of the most impacting form of information for all generations, from middle-aged demographics to young adults, to even children.
Nowadays, most of the population extracts their knowledge through social media. In this context, the creation of automated accounts on these platforms can impact the perception of people, spreading false information or polarising content, and causing a clash between two sides not willing to discuss and find a middle ground. Bots can also post and interact with other accounts, and show a different distribution of opinions with respect to the real one, causing the public perception to be shifted, and inducing entities like companies or public organisations to move in certain directions because of the feedback given by them.
For these reasons, this project's objective was to observe the behaviour of bots and human accounts to find a way to detect patterns that characterise one's relations. The focus were the relations between the accounts, in the form of the "follow" action. This action is performed from one account to another, signalling that the user is interested in receiving updates on the interactions of that account, and it can be reciprocated.

## 3 Dataset

The analysis was conducted using the TwiBot-22 dataset [1], which is publicly available on GitHub[1]. TwiBot-22 is a comprehensive, graph-based benchmark for Twitter bot detection, featuring the largest dataset available up to date. It offers a diverse range of entities and relationships within the Twitter network, and boasts significantly improved annotation quality

[1] github.com/LuoUndergradXJTU/TwiBot-22

compared to previous datasets.
Given the extremely large size of the dataset, it was impossible to work with the full dataset with the available resources. For this reason, preprocessing has been used to elaborate the dataset more efficiently.

## 3.1 Preprocessing

In order to preprocess the data, two subsections were considered: tweets and edges. From the latter, the 'follower' and 'following' relationships were extracted and merged into the *follower—followed* relationship. This procedure was applied for every user in order to build the network and to obtain the number of tweets for each user. Then, to allow their storage in the available RAMs, chunks for edges and for tweets were created.
After this technical solution, to be able to work with all the data, the next step was the creation of the communities. The communities are sub-networks of people who used the same hashtag in their tweets. They will be exploited to check if the patterns found in the network are robust to network changes or are network-specific. To test the robustness of the patterns and to verify the results on different scales, the subnetworks were divided into different categories –large, medium and small communities.

## 3.2 Analyzed networks

As discussed above, subnetworks were divided into three classes[2], based on the number of nodes (i.e., users) they contain: small networks have less than 1000 nodes, medium-sized have between 1000 and 10000, and large have more than 10000. Among each class, the focus was put on communities which may have presented more polarising opinions, as they discussed hot topics at the time of the dataset creation.

- Large → ukraine, ai, covid
- Medium → nato, deeplearning, nftcommunity
- Small → ruleoflaw, feminist, agenda2030

As a result of the relation-merging process described in Section 3.1, the networks are directed and their shape is monomodal. In particular, an edge from node A to node B means that user A follows user B.
Figure 1 illustrates the feminist subnetwork[3], where node size indicates the number of tweets authored by each account. Nodes representing bots are coloured in red, while human users are shown in green.

[2] An interactive graphical version of the subnetworks is available here:
alessiacrimaldi.github.io/sna_project
[3] Available at:
alessiacrimaldi.github.io/sna_project/results/subnetworks/feminist/feminist_network

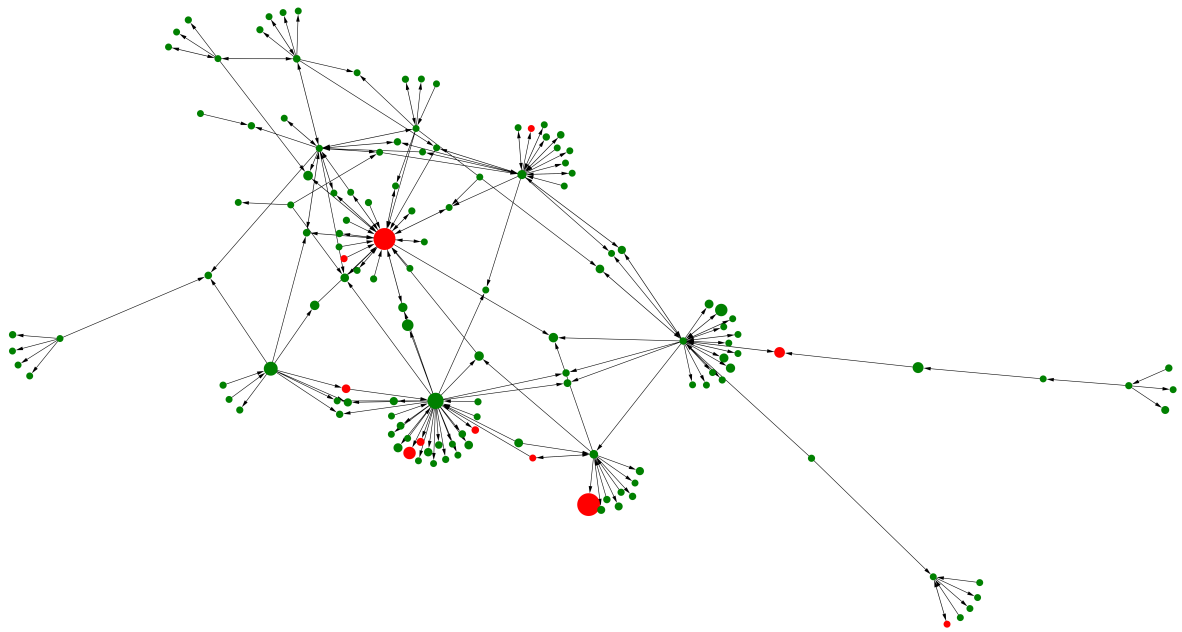


Figure 1: Feminist subnetwork

### 3.3 Adopted tools

The Python libraries used in this project were:

- The *Polars* [2] and *pandas* [3] libraries, and the *ijson* parser, for the manipulation of the dataset.
- The *NetworkX* [4] and *NetworKit* [5] libraries, to create the network and to calculate the measures on it.
- The *seaborn* [6], *SciPy* [7] and *scikit-learn* [8] libraries, to search for possible patterns in the bots/humans behaviour.

## 4 Validity and Reliability

As discussed in its official paper [1], the TwiBot-22 dataset was created trying to address and mitigate known problems of previous datasets, such as poor annotation quality and low dataset scale. This led to the construction of a large social graph with real world tweets, relationships between entities and metadata. This design allows the results to be statistically relevant, and it accurately reflects Twitter's social dynamics. Moreover, the dataset benefits from a strong annotation pipeline, which guarantees reliability and consistency, as well as reproducibility for a wide range of bot detection and behaviour analysis studies. It is also easily accessible and freely available.
In this work, the dataset was exploited focusing only on *follower–followed* relationships between users. Additionally, experiments were made by applying the same analytical measures to the full graph and subgraphs based on shared hashtags, in order to introduce a topical dimension to the analysis. This approach maintains the validity of the dataset, as both the follower links and the shared hashtag activity represent authentic user behaviours and capture meaningful patterns within Twitter's social structure. The results are also fully reproducible and

reliable, as a detailed description of the preprocessing pipeline, used to tailor the dataset to the specific research goals, is provided.

# 5 Measures and Results

As previously discussed in Section 3, during the initial experiments, several structural measures were computed on sub-sampled versions of the network to reduce computational complexity. However, the patterns that were intended to be identified did not clearly emerge. This has led to consider the possibility that this sub-sampling method might be introducing biases, potentially distorting or even removing the effects that could give rise to such patterns.
To investigate this hypothesis, the results obtained from the sub-samples were compared with those computed on the full graph. All experiments were conducted on an NVIDIA GeForce RTX 3090 GPU (24GB VRAM), which allowed to perform the necessary computations at scale. By directly analyzing the complete dataset, it was observed that the results were consistent with most of those from the sub-samples, suggesting that the sub-sampling procedure did not introduce significant bias. This confirmation supports the reliability of the methodology and validates the interpretability of earlier findings based on sub-sampled data.

## 5.1 Measuring the Network

One of the core assumptions in the study was the idea that bots show different behaviour when it comes to forming *follower–followed* relationships with other users. Especially, bots were expected to form less homogeneous connections, more based on random choices with respect to humans. As a consequence, the presence of bots was expected to have an effect on the overall network.
As a preliminary examination, to confirm previous suppositions on bot behaviour, it was decided to conduct some measurements on the entire network. Given the size of the dataset, the feasibility of some of these measures was conditioned by the computational limitations.
To highlight the effects of bots on the network, three main versions of the original *follower–followed* graph were created:

- original graph, with 693.761 users;
- human nodes graph, obtained by considering only human users, having 612.329 nodes;
- human-human edges graph, obtained by considering only human users connected only to other human users, with 589.924 nodes.

A key difference between the second and third subgraphs is that the latter excludes all those human users who were following or were followed by bots: a total of 22.405 users.
The analysis proceeded by examining the number of components present in each of these graphs, discovering that there seemed to be only one main large component which constitutes most of the graph, both in the first and third cases. As expected, the second subgraph presented a high number of isolated users.
The next step was verifying the scale-free nature of these networks. To do so, the degree centrality of all nodes was computed and their distribution was tested, fitting a power law and observing the results. They were positive, since all the three of them fitted perfectly in a power-law distribution, having $2 < \alpha < 3$. This is a further confirmation of the resilience of these types of networks to disruption and scale change, which should prove helpful for sub-sampling

in the next steps.
Finally, the hypothesis on bot behaviour was tested by computing density and clustering coefficient on the networks with and without bots. The findings are shown in Table 1.

| Network | Density | Clustering coefficient |
|---|---|---|
| Complete | $7.71 \times 10^{-6}$ | 0.06845 |
| Human nodes | $8.49 \times 10^{-6}$ | 0.06849 |
| Human edges | $9.141 \times 10^{-6}$ | 0.07109 |

Table 1: Network measures

It is observable that the human-human network has a higher density and clustering coefficient, which can be taken as a hint that bots tend to form fewer or sparser connections with their neighbours, thus partially validating the initial hypothesis. However, the small scale of the differences, considering the number of bots, reveals that their behaviour may be more difficult to spot than previously assumed.

## 5.2 Measuring the Nodes

Being tasked with finding notable features which may highlight the differences between human and bot users, it was decided to compute well-established network measurements on the graphs. Accordingly, from this point onwards, only networks containing both humans and bots were considered. As the objective of this project is to identify anomalies in the behaviour of specific users, the focus was placed on node measures. Alongside these, a few broader scope measures were added to show the presence of local groups around certain users.

**Centrality.**

- **Degree Centrality.** Considering the entire population of users, the degree centrality is the total sum of the number of followers a user has, and the number of accounts they follow. Moderate values might be expected for humans, with followers growing organically, while it is probably more common for bots to have extreme values, either high, if artificially inflated, or low, for simple spam bots. To better investigate this hypothesis, three different measures were considered: `in_degree` (the number of followers), `out_degree` (number of followed), `degree_centrality` (sum of the two).

- **Reputation.** On top of the simple degree centrality, the reputation measure was introduced as described in [9]. This acts as a ratio between the `in_degree` and the `degree_centrality`, and highlights how unbalanced the distribution between followers and followed is for each user. While most humans should have a ratio close to 0.5, some notable ones may be closer to 1 if they have proportionally more followers. On the other side, bots programmed to boost following were expected to have a reputation close to 0.

- **Reciprocity.** Leveraging the directed nature of the network, it is possible to compute how many of the follow relationships are reciprocated by each user. This may give an insight into the nature of users, as it is expected for most humans to follow each other back, while bots prefer one-sided relations.

- **Betweenness Centrality.** This measure allows to understand how much a node acts as a crossroad between paths from other users. This may be an index of how much a user

acts as a "common friend" between others. Higher values are expected for some humans acting as bridges between communities, while bots are more likely to be peripheral.

- **Eigenvector Centrality.** This centrality highlights the importance of a node depending on its neighbours. In this case, having a directed network, the NetworkX approach was followed by computing the left-eigenvector, which adds the centrality of the predecessors. This means that a node will be given more relevance if important nodes follow it. The expectation is: most bots have a low eigenvector centrality, as it is unlikely for them to gather too much legitimacy from human users; humans have a more balanced distribution, with some notables gaining more relevance.

- **PageRank.** Given that eigenvector centrality suffers from zero-trailing, it has been noticed that these types of networks contained a high number of elements with 0 `in_degree`, so it was decided to introduce the PageRank measure to investigate better the phenomena discussed before, accounting for the specifics of the problem.

- **Hubs & Authorities.** Another attempted approach to exploit the directed nature of the graph is the deployment of the HITS algorithm. Supposing bots are less followed than humans, it's unlikely for them to act as authorities and to be moderate hubs at best, while humans should have a more evenly distributed behaviour.

**Clustering Coefficient.** For each user $u$ and the set of its neighbours $N_u$ (users that follow it or are followed by it), this measure is the ratio between the number of couples of $N_u$ that have a relationship between each other and their total number. This gives an insight into how a user acts as a centre of their local community. It is supposed that humans are part of more meaningful communities, thus resulting in higher values.

**Average Neighbourhood Degree.** By examining the user's neighbourhood, the degree of their neighbours can be observed. Looking at the average of the neighbours' degree, it is possible to determine if the user tends to connect with nodes more integrated in the community, if they connect with users with similar integration or if the connections are not related to the importance in the network. This measure can also be used to observe the homogeneity of the network.

**Number of Triangles.** With this measure is possible to observe the number of strongly connected triples in the network. This is an indicator of the interconnectedness of the network, which presents strong and transitive relationships. Since usually humans tend to have "transitive" friendship relationships, creating triangles and strong connections, bots are supposed to be less integrated in the communities compared to humans.

**Core Number.** The *k-core number* is a measure that aims to observe how deep in the network a node is. The measure computes the value $k$ for which the node is part of a *k-core*. In this case, the core must be considered as a group of users, and it is possible to observe how much the user is entangled in the community considered.

**Number of Tweets.** This node attribute is the number of tweets posted by the user. It determines the "productivity" of a user in terms of tweets produced both in the whole dataset or in the considered community. In this project, this measure could be leveraged combined with other

factors to find users who are extremely passive or extremely active but isolated, behaviours typical of bots.

## 5.3 Distribution Analysis

After computing each one of the measures described above, it was necessary to examine their distribution across all the differently sampled subnetworks to observe if there were some interesting common trends among them, especially in the case of differences between humans and bots. To plot the distributions of each measure, it was chosen a simple histogram that compared the side-by-side values in each bin for bots and humans.
Considering the scale difference between the numbers of bots and humans, to highlight the eventual differences, density was plotted instead of the raw count. This way, the sum of the area of all columns will give 1.
Since most distributions presented a notable skew towards lower values, it was chosen to plot most of them in log-space along the x-axis to more evenly distribute the bins. Furthermore, to highlight their strictly tied connection, hubs and authorities scores were plotted together as an additional comparison.

## 5.4 Measures Discussion

After computing the measures described above for each of the proposed networks, they were analyzed by first observing the complete network results, then comparing them to the results shown by the subnetworks of different sizes. The goal here was to first detect recurring patterns in said distributions and to evaluate the effectiveness of each subnetwork at capturing the original structure.
Although small subnetworks have limited influence on the overall results comparison, they were included in the analysis for the sake of completeness and transparency. To improve readability, in some cases, the values of the selected measures for these smaller networks are graphically reported.

**Centrality.**

- **Degree Centrality.** From this simple measure, a slight difference is noticeable in the distributions of humans and bots, as the seconds seem to have a higher density towards zero (Figure 2), meaning there's a higher number of bots which have little to no connection, while humans tend to form more connections in general.
  Observing the `in_degree` (Figure 4) and `out_degree` (Figure 5) distributions, it's visible that this difference may be related to a higher number of incoming relationships with respect to the outgoing ones. This may suggest that humans have a higher number of followers.

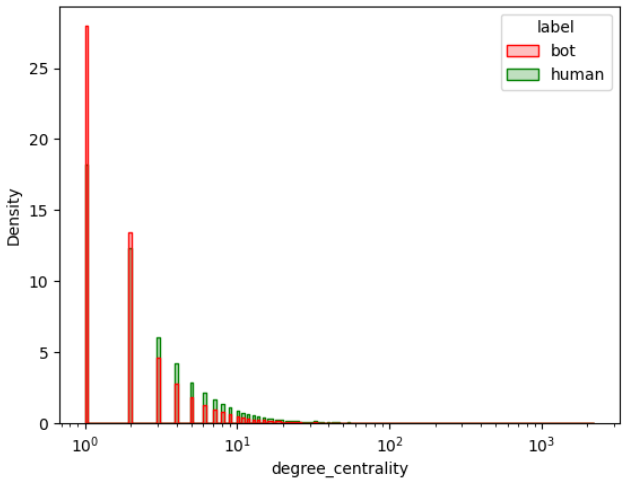



Figure 2: Degree Centrality distribution for complete network

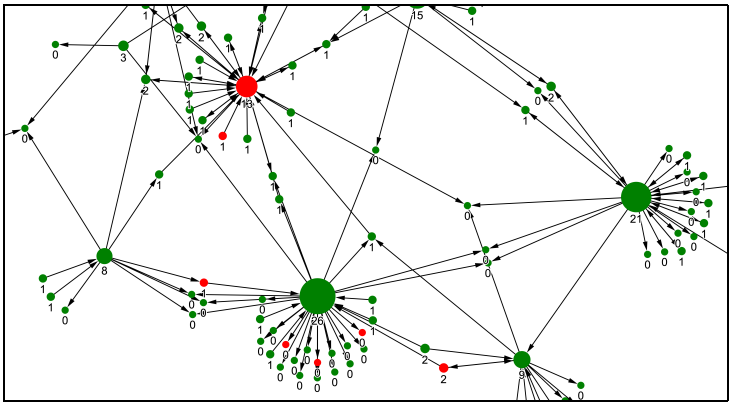


Figure 3: Degree Centrality values for feminist subnetwork

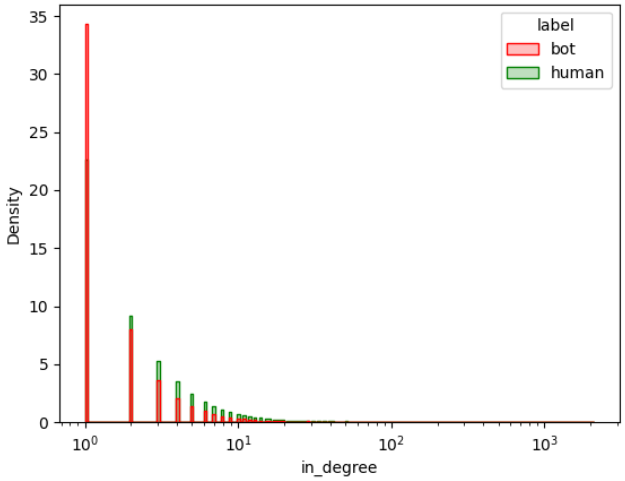



Figure 4: In-Degree Centrality distribution for complete network

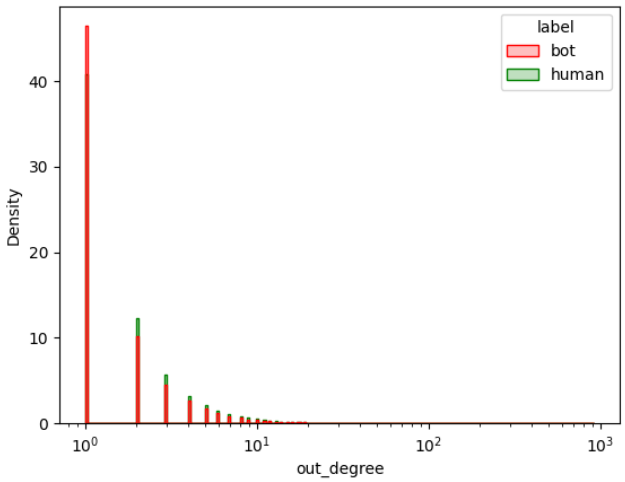
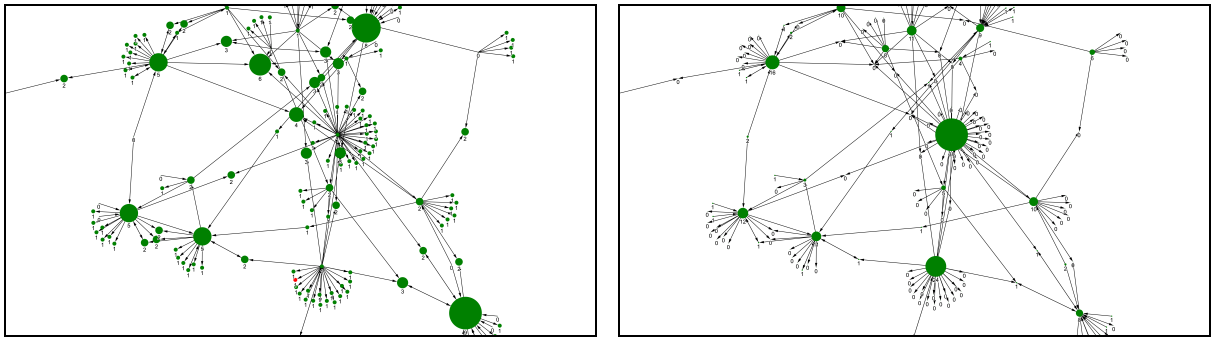



Figure 5: Out-Degree Centrality distribution for complete network



*In-Degree* *Out-Degree*

Figure 6: In-Degree and Out-Degree Centrality values for ruleoflaw subnetwork

- **Reputation.** Strictly tied to the degree centralities results are those from the reputation score, since they are derived from it. The obtained results match the predictions, as it is observable that the majority of humans fall in the central bins, having an average reputation score, with some notable ones having a very high score, close to 1. Concerning bots, a high concentration of low reputation scores can instead be noticed. Surprisingly, there is still a quite detectable number of bots with an average reputation score, but there are almost no bots with very high reputation scores.

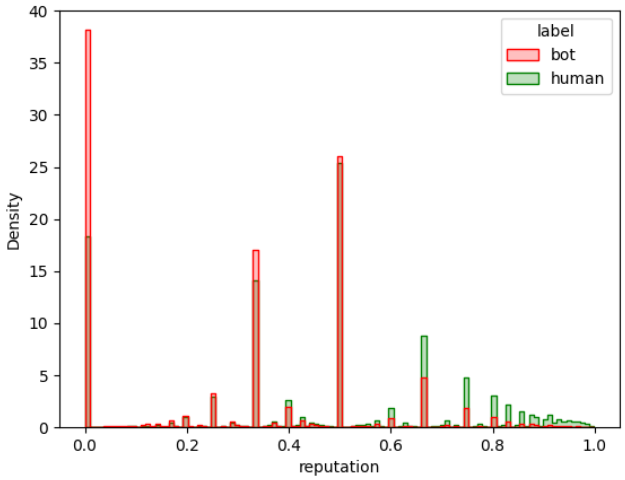



Figure 7: Reputation distribution for complete network

- **Reciprocity.** The results obtained for this measure are inconclusive. Even if the expectation was to find a bigger density of bot nodes near extreme values and a more homogeneous distribution for human users in the graphs, it can be observed that every distribution has his own patterns. As expected, the distributions of bots are usually skewed towards the extremes. However, in opposition with the initial beliefs, humans present a similar behaviour.

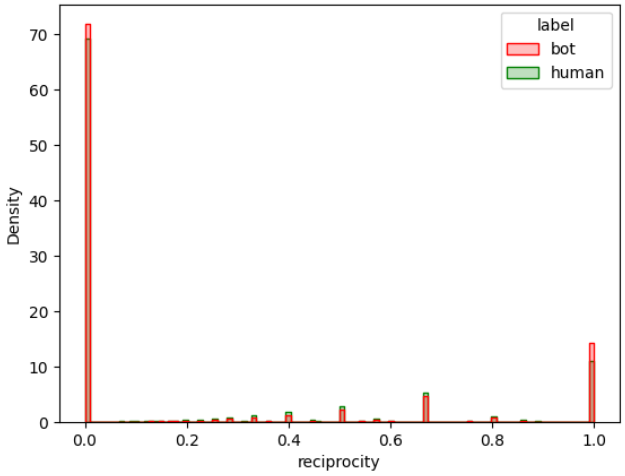



Figure 8: Reciprocity distribution for complete network

- **Betweenness Centrality.** In most cases, bots and humans exhibit very similar distributions of values, closely resembling the case of the complete graph shown in Figure 9. As expected, human data tends to be slightly shifted toward higher values.

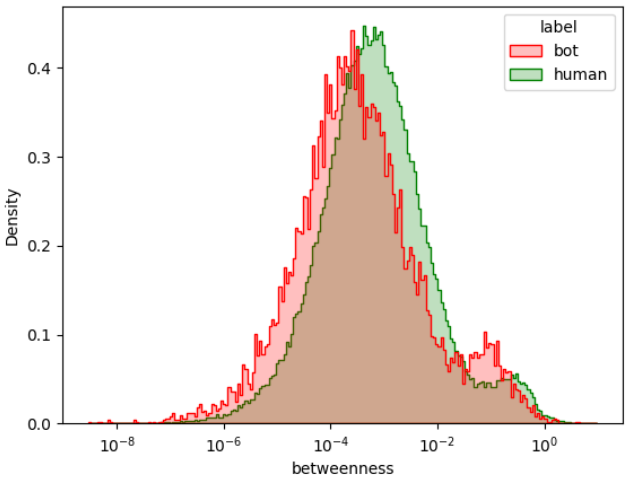



Figure 9: Betweenness Centrality distribution for complete network

- **Eigenvector Centrality.** As expected, in many of the analyzed networks, most bots exhibit extremely low values. However, the distribution for humans is not as balanced as anticipated and follows a trend very similar to that of the bots. Even at the highest value ranges, both humans and bots are present in most networks, contrary to initial expectations. Given the objective, the results obtained for this measure are not conclusive.

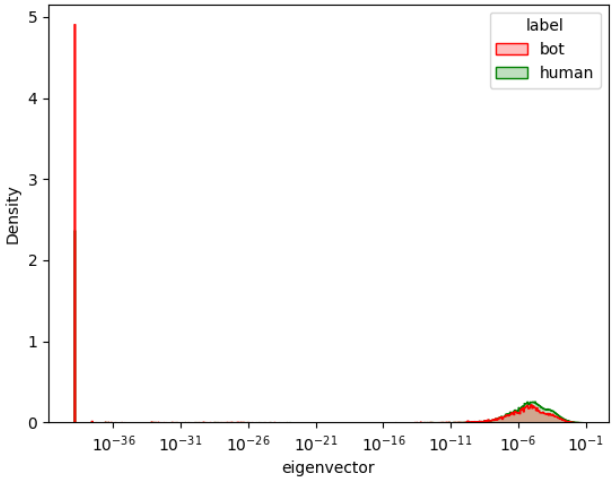



Figure 10: Eigenvector Centrality distribution for complete network

- **PageRank.** Being strictly related to the eigenvector centrality, the results for the PageRank are quite similar, with humans and bots showing comparable distributions of values in most of the considered networks.

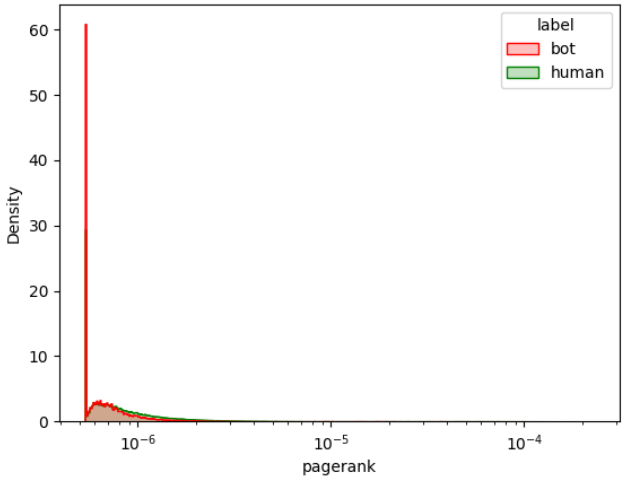



Figure 11: PageRank Centrality distribution for complete network

- **Hubs & Authorities.** The results are mostly inconclusive when analyzing hubs and authorities separately. For hub scores, humans and bots show almost the same distribution. As for authorities scores, in some networks, humans values are slightly shifted toward higher ranges, but not significantly, unlike what was predicted.
  Figure 15 shows that, as in most cases, bots are concentrated near the origin, with low hub and authority scores. Humans, on the other hand, are more uniformly distributed, covering a broader range of values for both hubs and authorities. This suggests that, while bots tend to exhibit low connectivity both as hubs and authorities, humans are more likely to participate in a variety of structural roles within the network. Nonetheless, the overlap between the two classes remains significant, particularly in the lower value region, which limits the discriminative power of these metrics.

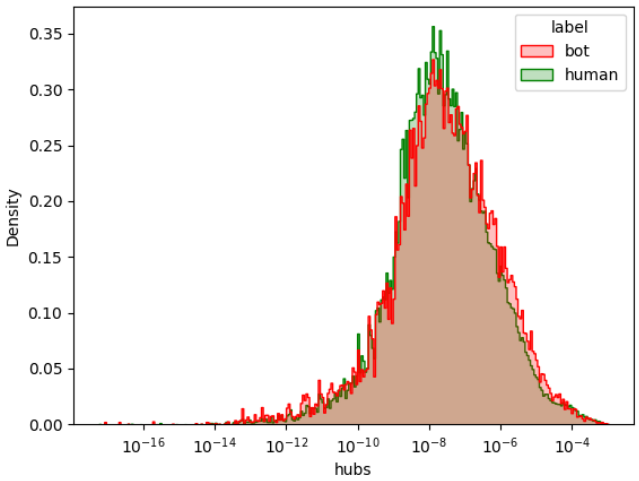



Figure 12: Hubs scores distribution for complete network

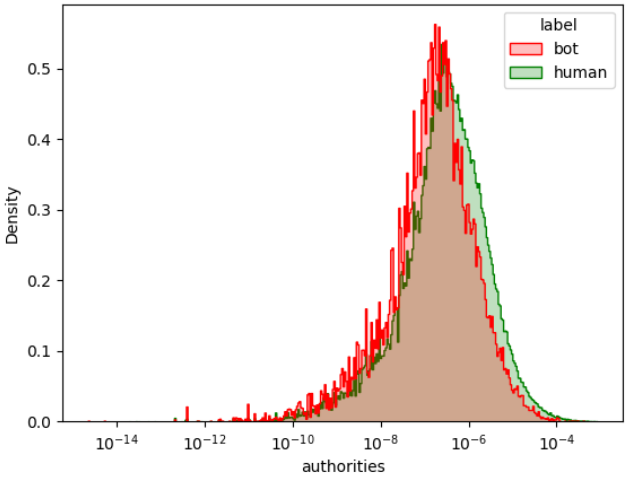
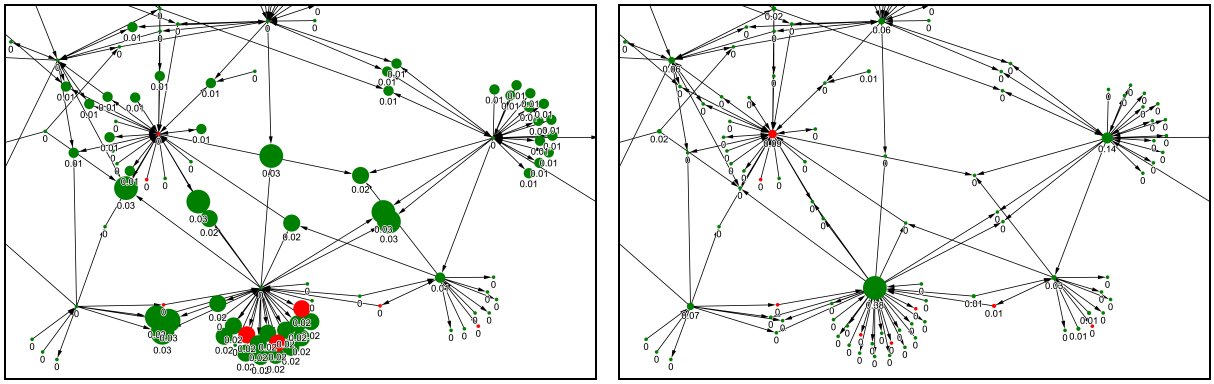



Figure 13: Authority scores distribution for complete network



*Authorities* *Hubs*

Figure 14: Authority and Hubs values for feminist subnetwork

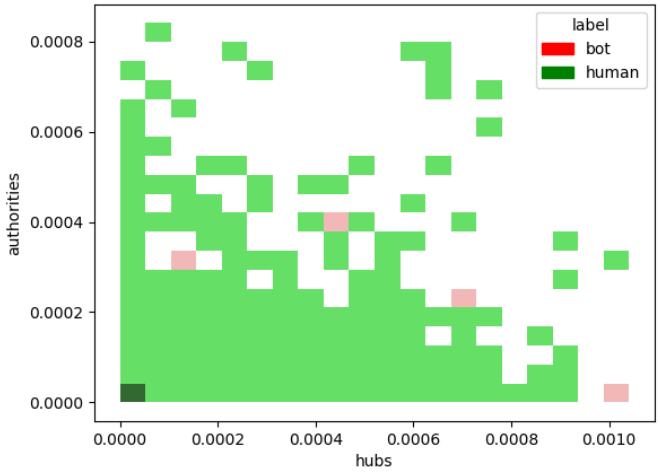



Figure 15: Hubs and Authorities joint distribution for complete network

**Clustering Coefficient.** Contrary to what expected, the clustering coefficient shows a higher density of bots at higher values. This phenomenon is likely due to bots with a reduced number of connections (even reciprocal, if they are part of an artificial botnet), which makes it easier for them to achieve high clustering coefficients. In contrast, human users typically exhibit a more diversified behaviour, resulting in more intermediate clustering values.

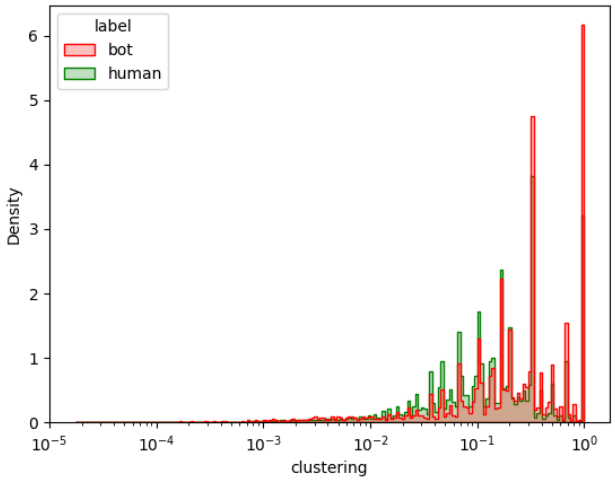



Figure 16: Clustering coefficients distribution for complete network

**Average Neighbourhood Degree.** Comparing the two distributions in the complete network, it is possible to observe that there is a slight difference between the two. While humans tend to concentrate more on the centre, giving rise to gaussian-like distribution, bots seem more skewed towards the lower end of the axes, thus showing a lower score on average. Another notable aspect is that, towards the right tail of the distribution, there seems to be another inversion as a higher concentration of bots with high average neighbourhood degree with respect to humans can be detected. From this single graph it can be inferred that humans have a tendency towards forming more evenly distributed connections. Unfortunately, these observations are not applicable to graphs obtained by the sub-sampled networks, thus reducing their effectiveness.

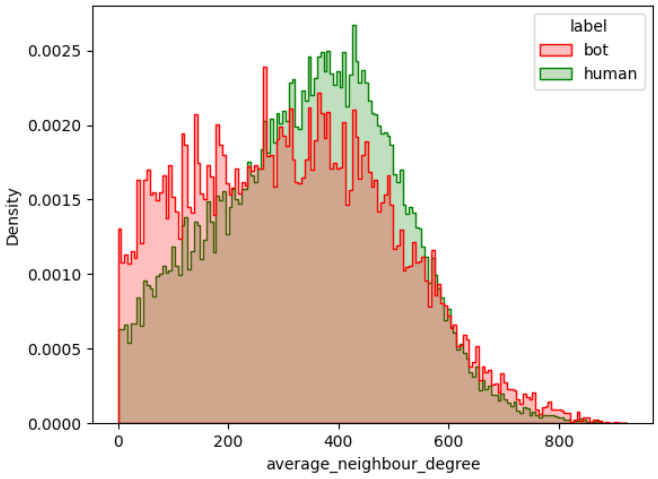



Figure 17: Average Neighbourhood Degree distribution for complete network

**Number of Triangles.** The objective of this measure was separate humans from bots under the assumption that humans formed more triangular relationships. This resulted inconclusive, since from all graphs it's possible to observe that the distributions for bots and humans overlap. The only noticeable difference is a slight tendency from bots in forming a few triangles, which weakly fits with the initial hypothesis.

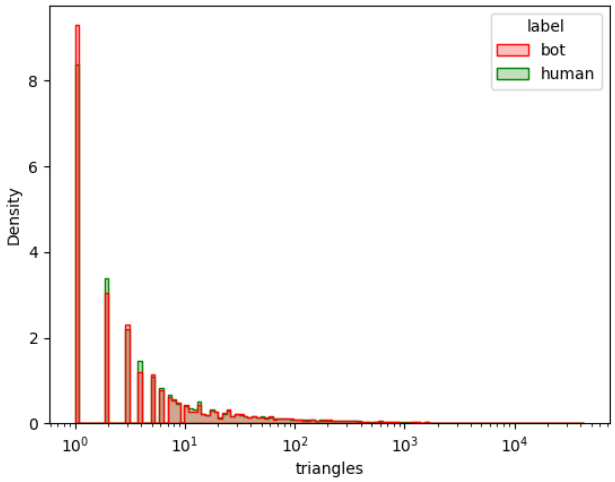



Figure 18: Number of Triangles distribution for complete network

**Core Number.** From the results obtained using this measure across all networks, bots appear to be more isolated, since the majority of them show a lower k number with respect to humans. This result signifies that bot users generally form more sparse groups, while humans develop more tightly connected groups. These results, again, do not show a strong tendency but highlight an interesting pattern worth mentioning, as they further confirm their trend towards isolation.

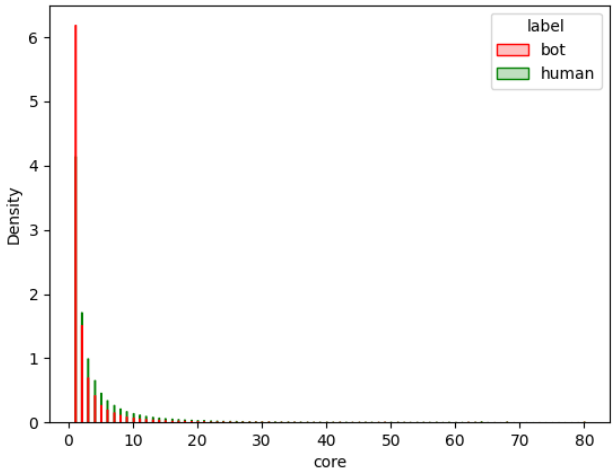



Figure 19: Core Number distribution for complete network

**Number of Tweets.** This last measure, while unrelated to the network structure, was supposed to give useful insight into user activity. Despite this hypothesis, the results were in-

conclusive, since the human and bot distributions appear similar and they show no particular divergence. The only noticeable aspect is a slightly higher concentration of bots with lower number of posts, a phenomenon which appears to be shared across most subnetworks.

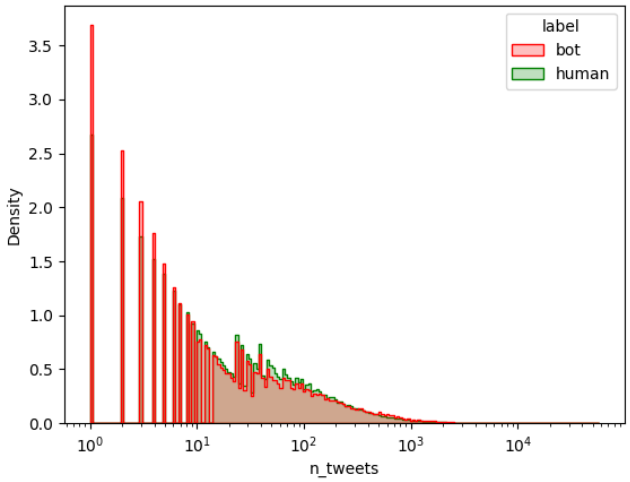



Figure 20: Number of Tweets distribution for complete network

## 5.5 Subsampling Results Discussion

To summarise the general outcomes of the subsampling process, it can be affirmed that from the obtained results a clear pattern emerged in the ability of gradually smaller subnetworks to capture the complexity of the original one. Large subnetworks seem to mirror almost exactly the original's behaviour. Medium-sized ones sometimes show signs of divergence, indicating some loss of structure or nuance. In contrast, small subnetworks seem to have a stochastic behaviour, being heavily influenced by the sampling process, only reflecting the most general trends.

## 5.6 Classification

To discover the most reliable relationships between the extracted measures and the labels (bot and human), it was decided to deploy three classical machine learning techniques for binary classification, as they would help highlighting the most relevant features while ignoring the least relevant ones. This activity was also necessary since no effective trend was noticed by examining each individual distribution across all measures.
As previously stated, three approaches were attempted:

- Naïve Bayes Classifier
- Random Forest Classifier
- Logistic Regressor Classifier

One of the tested aspects was the capability of subnetworks of capturing features of the complete network, so it was opted for a specialised pipeline for training and testing.
All measures were considered for the dataset, while the labels were used to indicate the target class. Firstly, all the measures obtained from one of the individual networks were used as bases

for the training and testing set, splitting them for 80% and 20% respectively. Secondly, each model was trained and evaluated using the described data.
To assess the generalisation capabilities of each network, a random sample of 5000 users measures, which were not included in the previous training and testing, was extracted from the complete network. The model was then tested on those.
Lastly, after observing the results of the previous experiment, these models were trained on the entire dataset. They were picked for their capability of handling large quantities of data. After an 80/20 train/test split on the measures of the full graph, training and evaluation were performed. To evaluate the results themselves, the accuracy and the F1 score of each prediction were computed, producing a confusion matrix for better visualisation.

## 5.7 Classification discussion

The results obtained from the classification experiments were underwhelming, especially those obtained using the subnetworks, since none of them was able to capture the complexity of the original one in a satisfying way. This problem became more severe the smaller the network used for training was, due to the reduced number of bots and the stochasticity featured by the measures, as highlighted in 5.5.
The following results contain an analysis of the performances achieved only by training on the entire network. Each experiment, while still failing to correctly detect the two classes of users, provides some interesting insights worth discussing.

**Naïve Bayes.** By observing the result of this classifier, it can be noticed that there is a high confusion between the two classes. This can be ascribed to the overlap present in all feature values. It is apparent that most humans end up classified as bots, and this can be attributed to the model assigning most of the uncertain cases to the bot class.

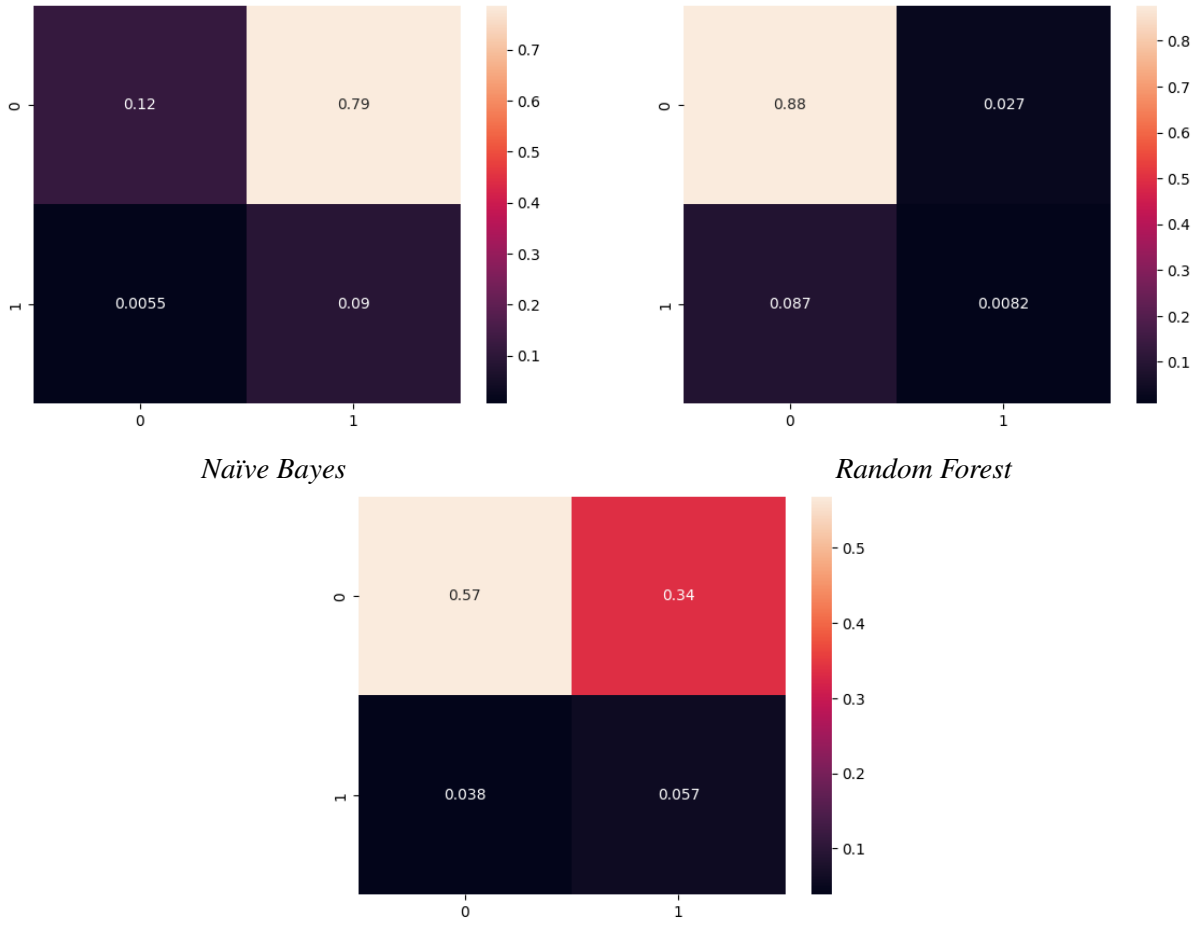
**Random Forest.** On the other extreme with the respect of the Naïve Bayes, there are the results presented by this classifier, since it manages to achieve a high accuracy by learning to classify most of the samples as humans. It disregarded the bot class, despite the usage of class weights to balance the distribution.

**Logistic Regressor.** Lastly, on the middle-ground between the other two, the results obtained from this model demonstrate an attempt at classifying the two classes. It achieved better results than a random choice, but ultimately failed due to the general overlap of the features. Here, most humans are correctly classified, but a high number was still marked as bots.

The values for accuracy scores for each model are shown in Table 2.

| | **Naïve Bayes** | **Random Forest** | **Logistic Regressor** |
|---|---|---|---|
| Accuracy | 0.21 | 0.89 | 0.63 |
| Macro F1 | 0.21 | 0.53 | 0.49 |

Table 2: Accuracy and F1 score for all models




*Naïve Bayes* *Random Forest*

*Logistic Regressor*

Figure 21: NB, RF, and LR confusion matrices (0: humans; 1: bots)

# 6 Conclusion

After conducting the experiments described above, it has been shown that, although some of the results could be interpreted as coherent with the initial hypotheses, they ultimately failed at solving the task of separating the two classes. The initial supposition was that the analysis of networks based on *follower–followed* relationships could provide an effective tool for bot detection.

By analysing the measures, some subtle patterns have emerged. However, none of them were individually effective enough for detection. Following these results, classical machine learning approaches have been attempted. Nonetheless, by considering multiple features, there was no improvement in the results.

# 7 Critique

It has been proven that the used *follower–followed* connection does not represent a meaningful relation for bot detection. This could be due to multiple reasons:

- The dataset was thought to be used in its entirety. Therefore, considering just a specific class of relationships greatly reduced its expressiveness.

- The measures applied could not provide enough significant information for the analysis. Perhaps, the use of measures tailored for this problem domain could have been more insightful.
- The deployed classification models had limited capability in understanding the detection task. Exploiting more complex architectures and algorithms may prove useful for tackling the problem. Among these, neural network architectures and deep learning models should be included, thanks to their ability to capture non-linear patterns.

Given these issues, a good approach for further development could be to consider relations based on users activity, since it is a more central feature aspect of Twitter. Among these, comments, replies, groups, and posts analysis, which were all included in the full dataset, should be considered.

However, it is worth highlighting that the majority of the simplifications made in this analysis were due to the limited computational resources available. Further development should account for larger computational power, especially RAMs, given the considerable memory requirements of the dataset.

# References


[1] Shangbin Feng, Zhaoxuan Tan, Herun Wan, Ningnan Wang, Zilong Chen, Binchi Zhang, Qinghua Zheng, Wenqian Zhang, Zhenyu Lei, Shujie Yang, Xinshun Feng, Qingyue Zhang, Hongrui Wang, Yuhan Liu, Yuyang Bai, Heng Wang, Zijian Cai, Yanbo Wang, Lijing Zheng, Zihan Ma, Jundong Li, and Minnan Luo. Twibot-22: Towards graph-based twitter bot detection, 2022.

[2] Ritchie Vink et al. Polars: Fast dataframes in rust and python, 2020.

[3] The pandas development team. pandas-dev/pandas: Pandas, February 2020.

[4] Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

[5] Christian L. Staudt, Aleksejs Sazonovs, and Henning Meyerhenke. NetworKit: A Tool Suite for Large-scale Complex Network Analysis. *Network Science*, 4(4), December 2016.

[6] Michael L. Waskom. seaborn: statistical data visualization. *Journal of Open Source Software*, 6(60):3021, 2021.

[7] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, B. Michel, V.and Thirion, O. Grisel, M. Blondel, R. Prettenhofer, P.and Weiss, V. Dubourg, J. Vanderplas, D. Passos, A. andCournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[9] Alex Hai Wang. Don't follow me: Spam detection in twitter. In *2010 International Conference on Security and Cryptography (SECRYPT)*, pages 1–10, 2010.

# A Github Repository

The project's source code and artefacts are available at the following link:
github.com/alessiacrimaldi/sna_project