HEC MONTRÉAL

Detecting Coordinated Activities through OnlyFans' Tweets

par

Maricarmen Arenas

Golnoosh Farnadi
HEC Montréal
Directrice de recherche

Sciences de la gestion
(Spécialisation Intelligence d'affaires)

Mémoire présenté en vue de l'obtention
du grade de maîtrise ès sciences
(M. Sc.)

December 2022
© Maricarmen Arenas, 2022

Abstract

Keywords

Research methods

In this thesis by articles, we present a research paper that we submitted for the WebSci'23 conference and is now under review. In addition to the article itself, in the thesis, we provide further detail regarding the motivation, background, literature review and research. The aim of this thesis is to provide a method that can facilitate the work of individuals combating online human trafficking. The majority of trafficking victims report being advertised online, this explains why online sex trafficking has been on the rise in the past few years. On the other hand, the use of OnlyFans as a platform for adult content has increased exponentially in the past three years, and Twitter has been its main advertising tool. Since we know that traffickers usually work within a network and control multiple victims, we suspect that there may be networks of traffickers promoting multiple OnlyFans accounts belonging to their victims. Based on these observations, we decided to conduct the first study looking at organized activities on Twitter through OnlyFans advertisements. Preliminary analysis of this space shows that most tweets related to OnlyFans contains generic text, making text-based methods less reliable. Instead, focusing on what ties the authors of these tweets together, we propose a novel method for uncovering coordinated networks of users based on their behaviour. Our method, called Multi-Level Clustering (**MLC**), combines two levels of clustering. In the first level, we detect communities based on

username Mentions and shared URLs, while the second level is done through two different approaches: i) the Partial Intersections (PI) of URLs and Mention communities ii) Joint Clustering (JT) by applying a subraph dense detection algorithm. We additionally successfully proved that our JT approach applied on synthetically generated data (with injected ground truth) shows a superior performance compared to competitive baselines. Furthermore, we apply the **MLC** to real-world data of tweets pertaining to OnlyFans and analyse the detected groups and show that our Partial Intersections provides good quality clusters (high entropy of OnlyFans accounts). Our paper and our thesis end with a discussion where we show carefully chosen examples of organized clusters and provide multiple interesting points that supports our method.

Contents

List of Tables

List of Figures

List of acronyms

ABBR Abréviation

BAA Baccalauréat en administration des affaires

DESS Diplôme d'études supérieures spécialisées

HEC Hautes études commerciales

MBA Maîtrise en administration des affaires

MSc Maîtrise

PhD Doctorat

This is dedicated to all human trafficking victims.

Acknowledgements

Professor Golnoosh Farnadi and her research team primarily work in subjects related to trustworthy machine learning, specifically fairness and privacy, therefore I am deeply grateful that she accepted to work with me knowing that my thesis is in another field (human trafficking detection). I'm also grateful that Professor Reihaneh Rabbany, an expert in the HT detection domain, accepted to co-supervise this thesis; her expertise was essential to the project. Furthermore, I want to thank both for giving a direction to this thesis – and paper – and for their continuous support, helpful discussions and their patience (specially during the summer when I encounter some medical difficulties).I also want to a big thank you to PHd student Pratheeksha Nair for collaborating with me and help pushing the submission of our research paper to WebSci'23, we couldn't have done it without her. Finally, I'm grateful for my supervisors and Mila for the financial support I received over the last 9 months; it has helped with financial stress and has provided me with the freedom to do better research.

Preface

Chapter 2 of this thesis has been submitted as a research paper and is under review in the WebSci'23 – 15th ACM Web Science Conference. This article is co-written with the supervisors, professors Reihaneh Rabbany (McGill University, Mila) and Golnoosh Farnadi (HEC Montreal, Mila), as well as PhD student Pratheeksha Nair (McGill University, Mila), who works on an adjacent area of analyzing ads posted on classified advertising websites.

The initial ideas of this research are from the supervsiors, which are then refined by Maricarmen after extensive exploratory analysis. She led the study, performed the domain and literature review, collected and pre-processed the Twitter data, which consisted of millions of tweets spreading over five years. Maricarmen also developed the data processing pipeline, and analyzed the results and case studies. Pratheeksha, contributed by running the baselines on the synthetic experiments for the paper, incorporating the joint clustering method, as well as helping in polishing the writing of the paper.

General Introduction

Sexual Human trafficking is a global problem that affects around 6.3 million people around the world, according to the Force labour and Force Marriage Report (**?**). In 2020, in the United States alone, there were 579 active human trafficking prosecutions; 94% of these prosecutions were sex trafficking cases. Out of the 1499 victims derived from these cases, 53% were children, mostly girls, and 44% of the victims were women. Over one third of the cases prosecuted were reported directly or indirectly by a victim (21).

Moreover, experts have demonstrated that the internet is tightly connected to the sex industry and that the material promoting human trafficking has increased dramatically over the years. In 2020, the primary method of solicitation in active sex trafficking cases, more than 80%, originated from the web. In fact, the internet has been the primary form of solicitation in sex trafficking schemes since 2008 (21). The COVID-19 pandemic has exacerbated this already monumental by reducing employment opportunities and isolating potential victims (8). In fact, low-income families and communities have faced even further economical instability while new precarious situations have risen for people who previously were living a comfortable life. Furthermore, stay at home orders have reduced the opportunities for employment for former victims of human trafficking (21). Among the potential victims, we account children and teenagers; the current climate has increased the likelihood of children and teenagers to fall victim to online recruitment and grooming. In fact, various countries reported an increase of commercial sexual exploitation and sex trafficking including child sexual exploitation material (CSEM) during COVID. In the Philippine, there was a 300% increase in referrals for potential online sex trafficking cases

from March to May 2020, including CSEM (21). In addition to this, because of the pandemic, planned anti-trafficking interventions have been put on hold.

Consequently, this situation has created new opportunities for human traffickers; they have adapted to the circumstances by further developing their online business model. For example, they have further developed non-physical methods of coercion to manipulate victims through the web; in the US, 75% of the prosecuted defendants controlled their victim by withholding pay (among other tactics) (24). Moreover, a survey ran by the Polaris project showed that 34% of their respondents, all OnlyFans creators, were controlled by their trafficker through their social media account. One of the methods of control, involved traffickers that would maintain fake accounts solely to bully their victims; keep them in fear (1). Considering that this report appeared in 2018, we therefore speculate that this type of tactics has evolved and increased. Another study shows that 56% of the sex trafficked victims do not advertise themselves, their handler does, and 17% percent of the victims say that they advertise themselves in conjunction to their trafficker (3). In other words, the victim's sex work is mainly advertised by a third party.

A few years ago, sites like *Backpage.com* and *Craigslist* were the main internet platforms reported for buyer solicitation in criminal human trafficking. Backpage.com specifically, was the biggest platform from 2010 until it was seized in 2018 (21). For this reason, much of the research made on human trafficking using machine learning, although not a massive amount overall, is based on data crawled from this specific site. These studies are focused on sex trafficking ads hidden among the 'regular' escort adverts and are aimed at easing the job of Law enforcement who usually rely on manual labour. Since Backpage.com has been seized, sex marketplaces have switch to similar websites such as leolist, which does receive a great amount of traffic. However, we can't discount the amount of promotion that is now also shared on social media and dating sites.

OnlyFans is the online platform that has disrupted the market the most in the last few years.OnlyFans is a subscription-based and pay per view (PPV) social platform that was created in 2016. The platforms host creators that offer live streams, chats and pornographic content (mostly) in form of videos and pictures, to to fans (subscribers) who pay

a monthly subscription or a one time fee to access the material (22). The price depends on the creator but OnlyFans keeps 20% of the revenue (4). What differentiates the platform from other pornographic site, is that content creators/sex workers create a sense of intimacy with their fans; in other words OnlyFans capitalizes on people's loneliness by mimicking intimacy and by showcasing specific fantasies and modes of communication. The constant interaction, which includes life streams, gives the fan the impression of living an intimate sexual relationship with the content creator (19). This is made evident by looking at the figures; in 2019, Onlyfans reached 8 million active users, in January 2020, it reached 12 million. By September 2021, this number increased exponentially; it reached around 170 million registered users (23; 4). On the other hand, there were 110,000 content creators in 2019, that figure nearly doubled by May 2020, while it attained 1 million by December 2020. As of September 2021, there were 1.5 million OnlyFans content creators. Note that most OnlyFans traffic comes mostly from the US – nearly 45% (23). Moreover, according to Google trends, OnlyFans search activity has doubled since the lockdown, and, according to Alexa, the website's traffic has exploded since (19). In other words, the pandemic created an opportunity for platforms such as OnlyFans. The platform has also grown thanks to their clever advertisement. It has built the reputation of bringing in high revenue to people without much effort; Onlyfans sells the dream of having to work less while acquiring more money than any other regular job, or while having a regular job on the side. In fact, this has attracted a crowd of people that would have never consider becoming sex workers prior to this; low wage workers and the unemployed (**?**). Unfortunately, many of the content creators realize that this is not exactly the reflection of their new reality; not everybody gets the chance to make money off the platform and being a content creator is more time consuming than expected. OnlyFans is thriving however, it has generated a great deal of money (2 USD billion in sales in 2020), and some creators have even made millions. The latest calculations made in 2020 related to creator's take home pay, however, shows that only 1% of them make 33% of the platform's revenue while 10% make 73% of that revenue. The median take home pay for a creator is $155 USD per month, and considering that creators have to be available 24/7

for their fans (posting frequently, promoting and responding DM's) (23), it's clear that it is a losing game for most creators. Nevertheless, we must bear in mind that OnlyFans is not only popular among content creators because of its earning potential, it's also because it's seen as a safer way to do sex work since the worker can bypass the potential violence related to the intimate physical contact with a prospect client. Moreover, it is also seen as an empowerment tool for some sex workers since they feel more in control of what they can do.

That being said, OnlyFans has shown multiple safety issues, the most serious one being catalyst for exploitation and criminal activity. A study made by the Avery Center notes that some creators continue doing in-person sex work or start doing in-person sex work after realizing that the revenue from the platform alone can't sustain them. In fact, forum chats indicate that many in-person sex buyers use the platform as a way of screening the person with whom they are thinking of buying sex from. That situation shows that OnlyFans is inadvertently cooperating to unsafe sex work (**?**). Even more alarming is that the Avery Center conducted a survey and established that 11% of the respondents were aware that minors had OnlyFans accounts and in conjunction, the participants were aware of 35 minors that sold material on the platform (**?**). In fact, OnlyFans doesn't have an effective way of screening users, thus minors can easily bypass the security measures put in place by the platform. Furthermore, in 2022, Pornhub, a pornographic site, caused outraged when it was made public they hosted videos containing sexual violence, nonconsensual pornography, and sex trafficking (5). All these videos where removed from the platform. Furthermore, Visa and MasterCard decided to stop processing payments from this site, which has caused concern among the small content creators of the platform (10). Consequently, opportunistic traffickers looking to drop illegal material as well as small content creators, are turning towards OnlyFans.

An important sign of human trafficking is when a victim is controlled by a third party. According to the Avery project, one third of the participants admitted to know whenever a content creator is managed by a trafficker, which implies that the phenomenon is not uncommon (**?**). In addition to this, law enforcement, federal agents and human

trafficking investigators have found signs of sex trafficking within Onlyfans. OnlyFans content is hidden behind a paywall, therefore social media has become the entry point to the platform. Hence, these professionals conducted their investigations through publicly available social media; they have analysed 97 public Instagram accounts. They concluded that thirty-six percent of them were likely controlled by a third party. Another issue of the platform is the 20% cut they take from the creators, this is a steep price to pay, specially if we consider that some of these workers have handlers who also get a cut from their revenues.

Online statistics show that 22% of the traffic comes from social media and the rest comes in from direct links (**?**). OnlyFans' social media promotion channel of predilection is Twitter. Nearly 10% of the traffic comes from that channel (4). Compared to other platforms, Twitter is lenient towards NSFW content or adult content(7). Moreover, OnlyFans offers Twitter integration; which allows users to automatically send posts created in the platform to Twitter (6). In fact, in our analysis we found out that much of the tweets were automated messages.

Moreover, from 2017 to 2021, Twitter saw the number of users increase by 97%, in other words, the platform is becoming more and more popular (2). Furthermore, as we can see in figure 2.1 from our paper (see section 2.3) that tweets containing the word OnlyFans increase between 2017 and 2021. These increases are most notable between years 2018 and 2019 (495%) and between 2019 and 2020 (260%).

Problem Formulation

We can summarize the problem as such: online sex trafficking has increased over the years and the pandemic has magnified the phenomenon in addition to having encouraged sex content platforms like OnlyFans to expand. The popularity of OnlyFans, has led the platform to be promoted on social media, mainly through Twitter . On the other hand, human traffickers have taken advantage of the situation and therefore developed new ways of trafficking their victims; one of these methods is through OnlyFans. We also know that

a big percentage of the promotion of human trafficked victims is done by a third-party entity, most likely a trafficker or a trafficking network. In other words, human trafficker are potentially trafficking victims on OnlyFans while promoting them on social media, mostly on Twitter.

Previous work on online HT has focused on ads rich in text (with distinctive information), consequently the research has focused on NLP and labelled data ads (15; 20; 17; 16). Preliminary research on our data has shown us that our tweets are composed of generic text, consequently, our work focuses on connections between the authors; groups of people advertising the same OnlyFans account. In other words, we want to find coordinated behaviour, also called lockstep behaviour. This is done by detecting abnormal high levels of synchronization between two users or more through their shared activity (shared hashtags, shared mentions, common urls, etc).

In fact, our paper is the first to curate and study an extensive dataset of tweets advertising OnlyFans accounts using an unsupervised multi-level approach (**MLC**) for detecting coordinated behaviour (lockstep). There is at least one study that links HT detection to lockstep behaviour (12). Most studies using lockstep methods are often applied on social media, specifically to find bot accounts, or suspicious networks of people (9; 11; 18). To conduct our research, we crawl around 2 million tweets between 2017 and 2021, mentioning the keyword OnlyFans. Every year is analysed separately. We start by uncovering (graph) communities of authors (nodes) through their shared Mentions and URLs posted. The second part of our method, is represented by two approaches : i) we find partial intersections between URL and Mentioned communities ii) we find Joint Clustering by applying a subraphs dense detection algorithm. Paper by (14) indicate that suspicious groups are present where there are high levels of similarities, therefore we specifically investigate small communities to find signs of trafficking. Moreover, there is a higher possibility of a sock-puppet account situation within small group of authors (also called Tiny clusters), this is when a user creates multiple false accounts (13). Consequently, we have a group of accounts, controlled by the same person, promoting a set of OnlyFans accounts.

In our paper, we showed through controlled synthetic experiments, that the performance of **MLC** is superior to other methods deemed comparable. Our Partial Intersection is poorly represented by our synthetic data (more explanations within the paper in Section 2.4.1 and in our general conclusion), therefore the performance of the latter should not be considered as representative of the method's capacity. However, we note the that PI approach is able to provide a better quality of clusters than the JT approach.

Contributions

In this paper, we have several contributions which are summarized as follow:

- We provide the first study analysing patterns of tweets used to advertise OnlyFans accounts as a proxy for understanding how sex work advertisements are moving towards social media platforms.
- We introduce an unsupervised method for detecting a specific type of lockstep behaviour in twitter that leads to the uncovering of organized groups.

Structure

Following the introduction, we present the literature review (see section 1.2.4) followed by the background (see Chapter 1), where we focus on the previously developed machine learning methods we employ in our work. We then present our paper (in Chapter 2), where we explain in detail our method and experiments. This is followed by a general conclusion (in section 2.7) here we discuss our overall work including the information provided in the Appendix .

Chapter 1

Background

In this chapter, we review the machine learning methods that our approach relies upon. For our proposed lockstep detection methodology, we employ the Louvain community detection algorithm (see Section 1.1.1), and various techniques to be build our Joint Clustering approach (see Section 1.1.2) as intermediate steps. Furthermore, the methods utilize to run our synthetic experiments are all listed in Section 1.1.3. We also performed peripheral work with K-means (see Section 1.2.2) using TF-IDF as text representation (see Section 1.2.1) as well as some tests using the Infoshield algorithm see Section 1.2.4. Below we have a short but comprehensive overview of each technique.

1.1 Main Method

Partial Intersection, our main method, which is defined in paragraph section 2.5.2 from section 2.4 in our the paper, is derived from the outcome generated by the Louvain community detection.

1.1.1 Louvain Community Detection

Community detection techniques are suited for social media since their goal is to detect communities of entities with common interests. Community detection method utilizes

graph representation; they work with nodes and edges. The nodes are the data points we want to create communities from, while the edges represent the interaction between the nodes. The Louvain method is a well-known community detection algorithm developed in 2008 in paper (3). The algorithm's speciality is detecting communities within large datasets by using an iterative process. Just like clustering, this method does not need labels; it is an unsupervised learning process. However, both methods have important differences. Clustering generates groups based on a specific set of features that characterize the data input, while community detection creates groups of nodes based on the strength of their interactions (edges), in other words, based on the structure of the network (28). The Louvain algorithm executes in two steps and are repeated iteratively, each iteration of the two step process produces a partition containing a set of communities. In short, a partition is a set of communities. Note that the creators of the method only validated the results of the last partition, which is also called "best partition".

The first step executed by the algorithm is called the modularity optimization while the second steps is called the community aggregation. Modularity of a partition can be defined by the formula . Modulatiry yields a scalar value between -1 and 1. This score can be interpreted as the density of edges within a community compared to the edges outside that community. The higher the score, the closest we are to fully modular clustering (3).

$$Q = \frac{1}{2m}\sum_{i,j}[A_{i,j} - \frac{k_i k_j}{2m}]\delta(c_i, c_j) \tag{1.1}$$

1. $A_{i,j}$ is the adjacency matrix, it represents the weight of the edges linking the nodes i and j
2. c_i is the community to which the node belongs to
3. k_i is the sum of the weights of the edges incident to node i,
4. $k_{i,in}$ is the sun of the weights of the links form i to nodes in C and m is the sum of the weights of all the links in the network

To achieve the first step, we start by assigning a community to every node in the network. Given a node i and its neighbors, j nodes , we calculate the gain in modularity if we

were to remove node i from community i and instead assign the node to community j. If the gain is positive then we move the node to community j, otherwise i stays in its original community (i). This step is applied repeatedly to every node in the network, meaning that it can be applied to the same node several times. Moreover, this step is applied sequentially, meaning that the nodes are processed in a certain order. Consequently, the order of the sequence can produce different outputs. The first step is only completed once the when a local maxima of the modularity is achieved; when we can no longer move a node without losing modularity (3).

The formula 1.2 below is the computation necessary to move an isolated node i to another community C, in other words, it is the formula necessary to calculate the gain in modularity ΔQ.

$$\Delta Q = [\frac{\sum_{in} + k_{i,in}}{2m} - (\frac{\sum_{tot} + k_i}{2m})^2] - [\frac{\sum_{in}}{2m} - (\frac{\sum_{tot}}{2m})^2 - (\frac{k_{in}}{2m})^2] \tag{1.2}$$

1. $\sum_{in}$ represents the weights of each edge inside community C
2. $\sum_{tot}$ sum of weights of edges incident to nodes in C
3. k_i is the sum of the weights of the edges incident to node i,
4. $k_{i,in}$ is the sun of the weights of the links form i to nodes in C and m is the sum of the weights of all the links in the network

In the second step, we build a new network from the communities created previously; the communities of nodes are turned into giant nodes. Here the he weighted edges (of the previous nodes) within these giant nodes are added to create the weight of the edges between each these giant nodes (24). The first phase is reapplied to this new network takes place only once the second phase is completed. The iterations through the two phases continue until maximum modularity is achieved. Note that each new network created after completing phases 1 and 2, constitutes a new level containing a new partition.

1.1.2 Joint Intersection clustering method

The Joint Intersection method was developed to add complexity to our work. This work was implemented by Pratheeksha Nair. The method uses various techniques to solve

the block-dense problem derived from our data, in the below paragraphs we explain the following methods employed: HDBSCAN, Dense Subgraphs and Singular Vector Decomposition.

HDBSCAN

HDBSCAN described by Campello et al. (5) is an extension of the Density-based clustering of applications with noise (11) (DBSCAN) algorithm. DBSCAN groups points together by measuring their distance and by establishing a minimum number of points. The points that are outside a density zone, are marked as outliers (10). HDBSCAN computes the hierarchical cluster tree of DBSCAN clusters, and then uses a stability-based flat cluster extraction (15). The steps go as follow (Github):

1. Transform space (the matrix data) based on its estimated density (agglomeration of points)
2. create a cluster hierarchy based on their level of density
3. truncate the tree of hierarchical cluster by removing the clusters containing less than the required points.
4. extract the clusters

Dense Subgraphs

A dense subgraph can be defined as a subgraph containing many connections between its nodes. This concept can be found in various type of networks including social networks. Formally, the density of a subgraph with k nodes can be computed by number of edges divided by the maximum possible number of edges (29).

Taken an undirected graph (this thesis is based on undirected graphs) $G = (V, E)$, then the density of a subgraph with vertex set S can be measured as such (18):

$$d_{(S)} = \frac{|E(S)|}{|S|} \tag{1.3}$$

where $E(S)$ stands for the set of edges in the the subgraph produced by nodes in S

Singular Vector Decomposition

Singular value decomposition (SVD), can be presented a matrix $M \in C^{m \times n}$ being decomposed into 3 matrices (22):

$$M = U\Sigma V^* \tag{1.4}$$

1. U is an unitary matrix $m \times m$
2. Σ is an $m \times n$ diagonal matrix
3. V is an $n \times n$ unitary matrix
4. V^* is the conjugate transpose of V.

1.1.3 Baseline Methods: synthetic experiments

Synthetic data experiments were run to uncover the groups injected in the synthetic graphs by using various methods. These methods were then compared to **MLC**. The methods below are used for the comparison, they include: Infomap, Node2Vec, Attri2vec, GraphSAGE and pcv. Note that one of the baseline are the original groups derived from the original communities we generated from Louvain.

Infomap

The Infomap algorithm (25) tries to compress the conditional information of a random walk on the network into Huffman codes. The algorithm tries to maximize the information of the random walk by capturing the conditional flow of information through the whole network. It also tries to find the partition that minimizes the description of any random walk on the network. The speed of the algorithm is comparable to Louvain (6).

node2vec

(13) propose the Node2Vec algorithm that maps nodes in a graph to an an embedding space that is of lower dimension, meaning that it contains in a smaller number of nodes

than the original Graph G. However, the algorithm still tries to preserve the same structure as the original Graph G. These embedding spaces are made of vectors corresponding to each node in the network. This algorithm can be used, among other things, for community detection (**?**).

attri2vec

Attri2vec proposed by (30) is an algorithm that performs a linear/non-linear mapping based on node content attributes to learn node representations. DeepWalk/Node2Vec are used to make the learned node representation respect structural similarity. Given a node pair (v_i, v_j) derived from a random walk, attri2vec uses a three-layered neural network to learn how to represent the target node v_i and using it to predict the existence of context node v_j. The model is trained end-to-end by minimizing the loss function regarding the predicted node pair labels and true node pair labels, which is achieve by applying stochastic gradient descent (SGD) updates of the model parameters.(7)

GraphSAGE

GraphSAGE by (14) is an algorithm principally used for graphs with rich node attribute information. It generates low-dimensional vector representation for nodes. Unlike other frameworks, GraphSAGE uses an inductive approach (can be generalize) that helps efficiently generate representations on previously unseen data (21).

PCV

PVC,a method developed by (23) is a simple 2 step algorithm that successfully discovers the ground truth clusters in bipartite graphs based on a standard graph model. A bipartite graph is a graph that is divided in two independent sets (here the sets are visualized as set on the right and set on the left). The nodes within the same set are not connected with each other. The algorithm's two steps are divided as follow : i) clusters the nodes on the left side of the graph based on the similarities of the neighbouring nodes ii) Clusters the

nodes on the right side of the graph based on the results of the left side of the graph using a degree-threshold. This algorithm is superior to similar algorithms since it also uncovers clusters of size $O(n^{\varepsilon})$ or 'tiny clusters', and that even with high destructive noise - here n stands for the number of nodes on the right side of the graph and $\varepsilon > 0$.

1.2 Methods for peripheral research : Appendix

Before establishing which method we wanted to implement for this thesis and paper, we analysed our data. This analysis was comprise of various steps, including NLP analysis. Here below we explain the algorithms behind KMEANS and Infoshield methods, which are the specific techniques we used during our analysis.

1.2.1 TF-IDF

K-means clustering on textual data needs to be transform into data the computer can understand, in other words we perform word embedding before applying K-means. TFIDF is a frequent and simple word embedding method that helps to determine the mathematical significance of words in documents using a statistical measure (1). The output is a vector made of numerical values that can now be read by the computer for clustering. TF-IDF stands for term frequency – inverse document frequency. The term frequency is simply the number of instances a specific term appears in each document over the total number of terms present in the said document. The total number of terms from all documents can be called the Bag of words (BOW), if for example, the BOW contains 20 terms then the TF vector per document will have a vector of 20 dimensions (4). If a document doesn't contain a certain word, then the numerator is zero. IDF, on the other hand, helps us measure the uniqueness of a term versus to a corpus of documents. This means that if a word appears on multiple documents and on our target document, then the word is not special and shouldn't get a high value. Mathematically, IDF is the log of the number of documents on the number of documents containing a specific term. The log is smaller if

the fraction is smaller. For example, if the term 'the' appears in all documents then it's IDF value will be close to zero. It's important to add that common words like 'the' are usually removed from the whole corpus before starting the TF-IDF process, in fact, in this research that word is removed using the 'stopwords' function from the nltk library (4). Note that other libraries and functions have also been used to understand the importance of words or terms in our data set (17). WordCloud, is a library that helps us see the importance of a word graphically. Counter is a function that returns the top words given in a given corpus.

1.2.2 K-means Clustering

Clustering helps us make sense of the data and its structure by creating subgroups within a corpus of data. Similar data are put under the same subgroup and data are represented by data points. K-means is an unsupervised learning method that can be based on different features. K-means uses centroids to define clusters (8). All the data points closer to that centroid are then part of the cluster, in other words, to establish if a data point pertains to a specific subgroup, the Euclidean distance between the data points and the centroid must be the smallest compared to the same distance from other centroids. The data points correspond to only one subgroup and the goal is to have clusters as homogeneous as possible. The K in K-Means refers to the number of clusters the user will assign to the process, this can be decided randomly or by using the Elbow method which is an heuristic method that plots all the values from a cost function given a different value for K (9). After specifying the number of clusters, the centroids are initialized. The centroid initialization is ,usually ,a random process that selects K data points for the centroids without replacement. The initialization is repeated until the centroid selection is optimized. The Euclidean distance is then calculated, and each data point is assigned to the closest centroid. The final step calculates the average of all data point per cluster to find the centroid location (8). K-means is an Expectation Maximization (EM) problem algorithm. The Expectation (E) part is trying to assign the data points to the closest cluster while the Maximization (M)

part is trying to find the maximum distance between the cluster's centroids (8). The following equation is our objective function:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} ||x^i - \mu_k||^2 \tag{1.5}$$

If datapoint x^i belongs to cluster k, then $w_{ik} = 1$, otherwise $w_{ik} = 0$, μ_k is the centroid of datapoint x^i.

This problem is in fact minimized in two parts, the first part 1.6 minimizes J w.r.t w_{ik} while μ_k stays fix. In this step, we calculate the euclidean distance between every cluster and a given datapoint x_i, we then assign that datapoint to the closest cluster. The second part minimizes J w.r.t. μ_k while w_{ik} stays fix 1.7. This step recalculates the centroids based on the new datapoints assigned (8).

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} ||x^i - \mu_k||^2 \Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = argmin_j ||x - \mu_j||^2 \\ 0 & \text{otherwise.} \end{cases} \tag{1.6}$$

$$\begin{aligned} \frac{\partial J}{\partial u_{ik}} &= 2 \sum_{i=1}^{m} w_{ik}(x^i - \mu_k) = 0 \\ &\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik}^{x^i}}{\sum_{i=1}^{m} w_{ik}} \end{aligned} \tag{1.7}$$

1.2.3 Clustering Implementation

To implement the K-means algorithm, we use the Sklearn library which offers the K-means function. The same situation applies to the TFIDF (8). We limit TFIDF feature vector to 2000 features which helps the processing time by discarding words that are less important. Stop words were already removed from our corpus. Furthermore, PCA is used to reduce the dimensionality of our feature matrix in two dimensions for visualization purposes. K-means is then applied using 5 clusters. In fact, PCA , which stands for Principal Component Analysis, is popular among large data sets since it helps reduce the dimensionality of the data, this is performed by transforming large sets of variables into a

smaller ones, while preserving the most important information of the sets (16). Note that we didn't employ the Elbow method, in fact, the number of clusters was chosen arbitrarly, as to see and compare the clusters from one year to another.

1.2.4 Infoshield

MSA and POA

MSA stands for multiple sequence alignment, while POA stands for Partial Order Alignment, the latter was developed by (19) and unlike earlier work on MSA, it uses partial order graph instead of profile sequence. MSA is an area with applications in DNA sequencing, but it is also present in the Natural language Processing sphere. In fact, paper (2) leverages MSA to learn the patterns of word sequences, while (26) uses MSA to align sentences by syntactic features.

Infoshield method

Infoshield is a novel algorithm that was introduced in 2021 by a group of Mila researchers working to combat online human trafficking through machine learning methods. The algorithm accomplishes the task by detecting near duplicate documents in a pool of documents. The algorithm not only detects these duplicates and assign them to a cluster, no matter how small the cluster is, but it also generates a template based on the meaningful documents within each cluster. Here, it is theorized that one person, a trafficker, is writing the ads for several victims, therefore the ads will be very similar. The method, according to the authors, is even more interesting when applied to smaller clusters. In addition to this, the algorithm helps detect bots by identifying exact or nearly exact ads and works well with tweets. Infoshield is divided into two main steps that can run separately. First off, there is InfoShield-Coarse which starts by taking the text and calculating the TF-IDF weight of every sentence that are up to 5-gram long. For further explanations concerning the TF-IDF method, see Section 1.2.1. Next, the sentences with low TF-IDF are eliminated and only the ones with highest scores are kept; the unimportant sentences are elim-

inated while the important ones are kept. The following step is the clustering, the process is based on documents sharing similar sentences. According to the authors, InfoShield-Coarse is too permissive, thus they present us with a second phase: InfoShield-Fine(see (20)). Infoshield-Fine is applied on the clusters produced by Inforshield-Coarse. This step generates templates and produces a word document for each one of them. The process is implemented in four phases, the first phase identifies a candidate set of documents which are represented by D_i, then, the set of documents is aligned using a Multiple Sequence Alignment Method (MSA) (see Section 1.2.4) for explanations on this technique. The authors chose the Partial Order alignment (POA) as their MSA method, the output of this process produces alignment A_i. Next, in the second phase, also called the *consensus alignment*, the algorithm decides which parts of the template must be kept, there should be a specific number of words. This problematic is turned into an optimization problem; the goal here is to find the best threshold h_i^* that will generate the best alignment at the lowest cost C (20).The optimization problem is represented by the following equation: (27)

$$h_i^* = min_h C(D_i|Sel(A_i,h)) \tag{1.8}$$

A_i represents the alignment, while D_i represents a set of candidate documents, and C is the cost. The function $Sel(A_i,h)$ will help find the best sub-alignment from the POA graph, here, we only keep the edges between words that occur more than h times in A_i (20). The results form the *consensus document*, in other words: the template. However, this template is formed of one sequence, meaning that there are no spaces for the insertions/deletions/substitutions. In the third phase, also named *the slot detection*, the algorithm finds the missing slots by computing the total cost of keeping or discarding a slot; only the ones that decrease the total cost are stored in the template. The final phase, also called *the relative length*, studies the clusters to see if the templates included are near duplicate or not and then assigns them a score. It is important to note that in our thesis, we only need to apply the Infoshield-Coarse since the output easily complements the results, we retrieved from Louvain. In fact, as we will discuss in our methodology and experi-

ment's sections, the data collected contains multiple sets of duplicates or near duplicates, therefore, we concluded that Infoshield method could be leverage.

Literature review

The presence of online human trafficking has greatly increased in the past few years, thus the phenomenon has caught the attention of researchers interested in solving this societal problem by using technological methods such as machine learning. Most of the literature is focused on online escort ads and sites such as backpage.com are the primary source of data. On the other hand, little research has been done related to social media ads. This is hardly surprising since social media is an even newer phenomenon.

Natural Language Processing and Human Trafficking: Most research papers make use of online escort ads as their main data source and employ NLP models as their main method. (8) leverages sentiment analysis by processing natural language text to generate classifiers for each ad (8). The authors establish a relationship between the 'love' label and ads they believe to be related to HT. This type of model has many limitations regarding the language. In fact, as mentioned in (16) traffickers know law enforcement is tracking their keywords, therefore they change them regularly. Furthermore, word ordering can be inconsistent and the use of emojis and symbols can further complicate the text. (16), use different methods to combat the linguistic problem by developing a language model that assesses the different issues that arise within a text in the context of human trafficking. In this paper the authors work with thousand of escort ads from the US and Canada, a subset of these ads is labelled with annotations divided in seven levels ranging from "Certainly no" to "Certainly yes". This type of annotation contrasts with the binary labelling and offers more nuance. Their model is a deep multi-modal network called the Human Trafficking Deep Network (HTDN). This model is divided in two pipelines used

simultaneously when processing ads: the linguistic network and the visual network. In regards to word embedding, which is related to the linguistic component, the authors use a skip-gram algorithm; this technique helps define words based on context instead of constituents. Furthermore, this method decipher symbols or slang that are usually ignored by regular methods. Sometimes, there is also irrelevant information within the ads. To combat this issue, the authors implement a time dependant embedding model at the word level (this is within their HTDN model). Thanks to this method, the model remembers the irrelevant parts, which means that if it encounters the same information once more, it does not have to reprocess it. The textual component of this method is best applied with ads that contain rich text, furthermore, it necessitates a certain amount of labelled ads, because our data (tweets) consist of mostly generic text, we can't rely on this type of method. In contrast (1) leverage a small set of labelled data and propagate the labelling to the rest of the data by using a semi-supervised framework S3VM-R. For this purpose, the authors collected 20k escort ads, from which they derived specific features that were used to keep the relevant ads. For example, if an ad contains text implying that there are more than one potential victim, the ad will then contain the feature 'Multiple victims advertised' and be added to the relevant ads. Results suggested that their approach is effective at identifying potential human trafficking related to online advertisements. This method limits the amount of labelling necessary, but still relies heavily in text that is rich enough to provide us with specific classifications.

Image Processing and Human Trafficking: In their paper (16) use images in the context of a vision network. As stated previously, their model HTDN employs both a visual and a linguistic network. The Visual network uses a deep convolutional neural network called Trafficking-VGG (T-VGG) which maps every image retrieved from the ads (or the user's description) and attaches a label to it.An end-to-end training is then performed. This model combines both a text and an image network and creates an explicit joint representation. The final result comes from the product of these two networks. The strength of this model lies on its dual components. (4) look for exploited minors within 55K tweets. Like the former paper, the authors use both NLP methods and computer

vision techniques. The visual component leverages the URL links extracted from the text. The algorithm tries to classify the gender and age group of the people present in the image by using SVM and Convolutional Neural Networks (CNN) techniques. SVM techniques produced a better output when classifying torsos and CNN performed better at identifying faces. A high percentage of the pictures did not show faces, consequently SVM classification was a better fit for this data in particular. Further studies on HT field should consider processing the images as well as the text; an opportunity is lost every time an image is discarded.

Social Media and Human Trafficking: As mentioned previously, studies that focus on detecting human trafficking using machine learning in social media are very rare. One interesting paper written on the subject is a thesis by a master's degree student published in 2021 . In paper (15), the author identified sex-industry posts to help support sex-trafficking investigators using a semi-supervised method. Using a set of sex working related hashtags as guideline, the author manually classified a subset of its 50,000 posts and labelled them 'sex industry' or 'not sex industry', while all the posts were embedded using FastText and Do2Vec. This labels were leveraged (label propagation) after performing K-MEANS clustering on the embedded posts. Furthermore, no expert is needed to label the data, we only need a set of keywords related to sex work. Our thesis is simplified by the presence of Twitter's API, therefore these steps are unnecessary, however it is important to know that we can build tools to filter this type of data from social media. As discussed previously, paper (4) employs both NLP and Computer vision methods. For the textual component of this, the authors leverage features related to HT (ex:third person use), and applied two Semi supervised techniques (SVM and Naïve Bayes) to classify tweets either as suspicious or non suspicious, the results are verified against labelled data annotated by experts. Both techniques yield good performance measures. This method shows that annotated labels are not necessary to detect HT in this type of data.

Big data and Human Trafficking: Authors in (11) were able to leverage massive deep web data; they crawled 14 million English escort ads using a novel machine learning framework to uncover commercial sex supply chains. first, the deep neural network is

trained so it can distinguishes recruitment from sales posts, it then extracts Meta data (phone number, location) from every post. This type of extraction is very similar to our feature extraction and reinforces the idea that this step should be considered.

Through their method, the authors are able to uncover 27 recruiting tactics by using very few labels.

The connection between the recruiting ads and the sex work is confirmed by the metadata. In fact, an important part of the paper is how the authors are able to make connections between seller and buyer using the metadata. By constructing recruitment-to-sales pathways, they are able to see patterns. In short, they are able to flag potential human trafficking by establishing connections between the deceptive recruiting ads and sex the sales ads. Here we clearly see there is some type of coordinated behaviour; the authors of deceptive ads share the same meta data as the sex sale ads.

Behavioural Analysis of Human Trafficking Ads: Other research have explored the idea of making connections between posts. (10) addresses this problem by using a method called Active Search of Connections which "finds related and relevant data points to a given lead through their shared evidence". This study created the RedThread model, which finds connections between ads using labels and evidences extracted from the data itself. For example, the label would be a phone number and the evidence will be its images, nickname, date and more, the model would be able to connect these data points and realize that the ads are posted by the same individual. ReadThread uses "Active search method" (10) which formulate the problem as a sequential Bayesian decision theory problem. It also uses exploration by inferring connections and finding relevant nodes. As a final step, the model develops a semi-supervised algorithms to cluster the heterogeneous data (reference). This research again focused on shared information, however they still rely on labels.

On the other hand, (6) presents a method entirely unsupervised that aims at finding cluster of texts that are nearly identical using a the Infoshield algorithm. The main goal of the paper is to identify human trafficking among escort ads and these templates help us see who is in charge of the same type of message; if an individual or entity is advertising

multiple people, the individuals involved can potentially be part of a trafficking network. (7) tries to detect illicit organizations among ads. For that purpose, the authors build and explore a template matching-based framework and solely using the information extracted from the data itself. HDBSCAN and the modified HDBSCAN with LSH algorithms are used to cluster the template, while graph based method is used to merge similar templates together. On the other hand, phone number matching is used as ground truth surrogate; the authors use the connected components of phone number co-occurrence network as stand-in for the organizations present in the data. The co-occurrence appears when two unique phone number both appear on the same ad. The results from the merged templates are then correlated to the surrogate phone number networks. The authors state that exploration of the results proposed gives a deeper insight into the human trafficking networks than results derived solely from phone numbers.

Lockstep Behaviour Detection: As mentioned previously (6) helps identify bot behavior on twitter, this is done by finding identical tweets or quasi identical tweets. Other studies have looked deeper into organized activity. In fact, lockstep behaviour for fraud detection has been used in multiple domains including bot and troll detection in online social media (9; 3; 12). In fact, papers like (9) employs an unsupervised method that detects coordinated networks using different shared features among accounts. The authors were able to find coordinated networks based on their usrname changes, image sharing, sequential use of hashtags, co-retweets, and synchronization. This paper proves that using features extracted from the paper itself can be effective in identifying lockstep behaviour on Twitter, and according to the authors the methods can be applied to other forms of social media. On the other hand, other papers such as (3; 2) use Lockstep methods to investigate fraudulent activity by detecting abnormally high levels of synchronize activity. (3) focuses on a method (ND-SYNC) that uses synchronized fraud detection via Retweets. The authors want to differentiate organic activity from spammy retweets originating from paid followers (which create false sense of popularity). The authors first extract features such as the inter-arrival time of retweets, the variance as well as the number of retweets within a retweet thread. They then analyse a set of retweet threads for

posts from a given user and compare it to the behavior of users within the same thread. Every user is then assigned a scored based on two specific concepts (“intra-synchronicity” and “inter-synchronicity”). Finally, the authors combine the score of every user from a specific thread, and report the groups of users that are the most suspicious. The method, which was applied on 12 million tweets shows an outstanding 97% accuracy. Furthermore, papers such as (5; 13; 14; 17) investigate abnormalities and fraud by leveraging scalable dense block detection techniques. In (5), the authors state that there are many false reviews on platforms such as Yelp, Amazon and TripAdvisor This is detrimental for the consumer and also for the company’s competitor. This is why this paper developed a novel approached called *FRAUDAR* that can be applied in real world graphs. Here the authors are looking for unusual dense and large regions in the adjacent matrix signaling a of fraudulent actions; the fraudsters have created too many edges. FRAUDAR is able to detect fraudsters that are under camouflage by employing different methods to make their strategies appear normal. According to the authors, these fraudsters seem to know the structure of the graph and that’s how they are able to hide from it. This statement triggers the question of whether or not the authors of OnlyFans tweets with nefarious intent, are able to hide and appear like any other regular user.

Chapter 2

Social Media as a Vector for Escort Ads: A Study on OnlyFans advertisements on Twitter

Abstract

The majority of trafficking victims report being advertised online, this explains why online sex trafficking has been on the rise in the past few years. On the other hand, the use of OnlyFans as a platform for adult content has increased exponentially in the past three years, and Twitter has been its main advertising tool. Since we know that traffickers usually work within a network and control multiple victims, we suspect that there may be networks of traffickers promoting multiple OnlyFans accounts belonging to their victims. Based on these observations, we decided to conduct the first study looking at organized activities on Twitter through OnlyFans advertisements. Preliminary analysis of this space shows that most tweets related to OnlyFans contains generic text, making text-based methods less reliable. Instead, focusing on what ties the authors of these tweets together, we propose a novel method for uncovering coordinated networks of users based on their behaviour. Our method, called Multi-Level Clustering (**MLC**), combines two lev-

els of clustering. In the first level, we detect communities based on username Mentions and shared URLs, while the second level is done through two different approaches: i) the Partial Intersections (PI) of URLs and Mention communities ii) Joint Clustering (JT) by applying a subraph dense detection algorithm. We additionally successfully proved that the JT approach applied on synthetically generated data (with injected ground truth) shows a superior performance compared to competitive baselines. Furthermore, we apply the **MLC** to real-world data of tweets pertaining to OnlyFans and analyse the detected groups and show that our Partial Intersection approach provides good quality clusters (high entropy of OnlyFans accounts). Finally, we discuss examples of organized clusters as case studies and provide interesting conclusions to our study.

clustering, community detection, dense block detection, social median networks

Social Media as a Vector for Escort Ads: A Study on OnlyFans advertisements on Twitter

Maricarmen Arenas, Pratheeksha Nair, Reihaneh Rabbany, Golnoosh Farnadi

June 28, 2023

2.1 Introduction

Forced sexual exploitation and Human Trafficking (HT) are nefarious crimes affecting over 6.3 million people in the world **?**). Technology is largely involved in the recruiting of HT victims MINOR (28) and a large number of these victims are advertised online on escort websites Polaris (30). This number has been growing over the years and the pandemic has magnified the phenomenon Adhoob (1) through its aftermaths of reduced employment opportunities leading to the isolation of potential victims Giammarinaro (11). Consequently, human traffickers have taken advantage of the situation and sought out new ways of advertising their victims[1]. We speculate that one of these ways is through OnlyFans, whose popularity has been in the rise especially during COVID-19 (26) and after the downfall of websites such as Pornhub (8) and Backpage (2). OnlyFans is a subscription-based and pay per view (PPV) social platform created in 2016 that predominantly hosts adult pornographic content. This platform has disrupted the market in the last few years

[1]https://www.state.gov/reports/2021-trafficking-in-persons-report/

and has seen an exponential increase in the number of users[2] (as made evident in Figure 2.1) especially since the pandemic (17). Users that post on OnlyFans also rely on external platforms such as twitter to promote their content, build communities and gain followers by sharing each others' posts (39; 40).

In this study, we investigate the interactions among tweets (such as mentions and external URLs) to help uncover coordinated and organized behaviour. HT is primarily an organized activity where one or more perpetrators traffic multiple victims. While this idea has been deeply explored for HT detection in escort ads (27; 38; 32; 31; 3; 24), there are no works, to the best of our knowledge, that look into similar behavior in social media networks. Our study is focused on identifying organized groups consisting of a network (or individual) advertising or promoting multiple OnlyFans accounts.

Drawing from previous studies that link HT detection to lockstep behaviour (22) and present evidence of lockstep in social media (13; 21; 36), we formulate this problem as one of identifying *tiny* clusters (relative to the size of social media networks). Intuitively, small groups of users regularly mentioning one other and referring to the same set of OnlyFans accounts could be sock-puppet accounts (23) handled by the same user and is what we are interested in discovering. It is to be noted that the detection of such groups warrants further investigation by human experts as identifying HT is not a trivial task. Furthermore, Paper (24) indicates that suspicious networks are present where there are high levels of similarities, therefore we specifically investigate Small communities to find signs of trafficking.

To this end, we present **MLC** an unsupervised multi-level clustering approach for identifying organized groups in Twitter. Our method processes both content (shared URLs) and connection (user mentions) information from the network in two levels by first detecting graph communities and then finding overlapping clusters. We present two approaches for the second part based on i) *partial intersections* and ii) *joint clustering*. We employ our method on controlled synthetic experiments (Section 2.5.1) as well as real-world Twitter data dated over five years (Section 2.5.2). Through the synthetic data

[2]https://www.followchain.org/onlyfans-statistics/

experiments, we demonstrate the superior performance of **MLC** in comparison to off-the-shelf contenders. We then provide an analysis of the clusters obtained from the real-world data. Lastly we also discuss three example clusters as case-studies.

The main contributions of this work may be summarized as follows:

- We conduct the first study analysing patterns of tweets used to advertise OnlyFans accounts as a proxy for understanding how sex work advertisements are moving towards social media platforms
- We introduce an unsupervised method for detecting a specific type of lockstep behaviour in Twitter that leads to the uncovering of organized groups.

The rest of this paper is organized into five sections. Section 2.2 covers the background and previous works related to HT and the detection of organized groups in social media. We introduce and discuss our real-world Twitter data collection in Section 2.3 and move on to our proposed method in Section 2.4. Section 2.5 elucidates our experiments on both synthetic and real-world data and presents our results. We discuss specific examples in Section 2.6 and conclude in Section 2.7.

2.2 Background

HT detection Online escort ads are the most commonly used data source in HT detection research and several methods have been employed to analyze them. These include supervised text processing models (27; 38; 3), unsupervised template detection methods (25; 24) and approaches that establish connections between authors using shared features such as phone numbers as evidences (31). Although we are still looking for connections between users, our work differs from these methods as we are focused on Twitter data which is much different from escort ads and contains connections among users. To the best of our knowledge, one of the only works that focuses on HT detection in social media(19) leverages a small set of labelled data and combines both image and text. This method also relies on tweets that are similar to escort ads and contain keywords related

to HT. In our case, we work with tweets related to OnlyFans promotions, which tend to utilize generic sexual vocabulary and therefore cannot be processed using this method; not to mention the lack of labelled data.

Dense block detection Lockstep detection is often applied on social media to find bot accounts (7) or suspicious networks of people (12). This is done by detecting abnormally high levels of synchronization between two or more users through their shared activity (hashtags, mentions, common URLs, etc). Dense block detection has also been used for finding lockstep behaviour in the context of anomaly and fraud detection (34; 35; 36). Finding cliques in graphs is also a related problem and most exact, heuristics and approximated solutions scale linearly with the size of the graph, making them difficult to work with on real-world data. Scalable solutions such as Frauder (20) and its extensions including offline (35; 36), online (37) and hierarchical (43) sub-tensor detection look for subgraphs with large average degree. These methods work on a single matrix (such as the graph adjacency matrix) whereas our method detects groups which enforce dense substructures in coupled matrices/graphs.

Graph clustering Dense subgraph detection can also be considered to be similar to graph clustering and community detection which finds groups of densely connected nodes. Louvain (5) and Infomap (33) are two common methods for community detection based on modularity maximization and information compression respectively. Community structure can also be recovered by clustering node embeddings which are essentially mappings of nodes and subgraphs to Euclidean spaces. Unsupervised GraphSAGE (18), node2vec (14) and attri2vec (42) are common network embedding methods and our synthetic experiments (Table 2.2) reveal that they are unable to recover tiny clusters as effectively as **MLC**. Moreover, none of these methods have been explored in the context of HT detection.

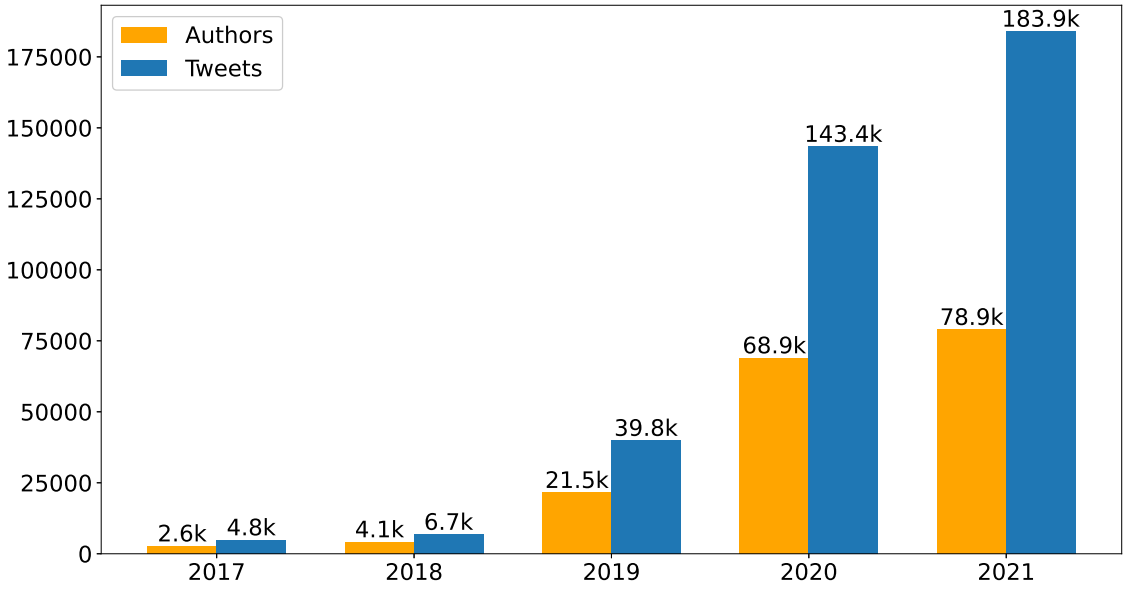


Figure 2.1: Growing popularity of OnlyFans in Twitter from the years 2017 to 2021 in terms of number of tweets and authors

Table 2.1: Dataset statistics

Year	# Tweets (RT)	# Authors	# URLs	# Mentions
2017	404,498 (57%)	63,181	120,495	8,075
2018	582,690 (71%)	96,265	148,878	13,138
2019	571,244 (88%)	137,823	204,068	26,201
2020	586,934 (91%)	192,991	375,788	58,910
2021	569,057 (94%)	194,235	369,706	64,379

Tiny cluster detection Identifying tiny clusters is very challenging and most modularity optimization methods depend on the network size and connectedness of clusters (10). pcv (29) is one of the few methods focused on finding tiny clusters with theoretical guarantees which considers bipartite stochastic block models. Our work considers a more general case of coupled matrices and such combination of different sources of information is proven to be a necessity for better recovery of community structure (9).

2.3 Data

We curated a dataset of 2 million English tweets related to OnlyFans posted between 2017 and 2021[3]. These tweets were extracted with the Academic track Twitter API (v2) using the query keyword of "OnlyFans". We then removed tweets containing the words 'promotion' and 'promote' as these are produced by individuals promoting their services to OnlyFans creators.

We collected tweets daily with a cap of 10,000 tweets per day (due to the API constraint of 10 million tweets per month). We started hitting this cap while collecting tweets from 2018 which means that the number of tweets mentioning OnlyFans in a single day crossed 10,000 sometime in 2018. In 2017 this number was much lower, which explains why the data collected from 2017 is significantly smaller in size, as seen in row 1 of Table 2.1. Although our method focuses on user mentions and URLs, we also collected other information on the tweets such as their retweets, authors, hashtags and geolocation (if available). Information on retweets (and replies) were crawled from and linked to their original tweets.

Additionally, we ran an experiment to confirm the growing popularity of OnlyFans in Twitter (as discussed in Section 2.1), which serves as the premise of our study. Using the same API, we crawled all available tweets mentioning OnlyFans posted on a single day (October 1st) of each year (2017-2021). Figure 2.1 shows a yearly increase in the number of tweets as well as the number of authors mentioning OnlyFans. There is a big jump from 2019 to 2020 with a 260% increase in the number of tweets and a 221% increase in the number of authors. It is also to be noted that Twitter itself saw a 97% increase in its users from 2017 to 2021[4]. Although we only analyzed tweets from one day of the year, it demonstrates the increase in OnlyFans related activity on Twitter which is further corroborated by other reports[2].

[3]We do not publish this dataset as it may contain sensitive and person-identifying information
[4]https://www.omnicoreagency.com/twitter-statistics/

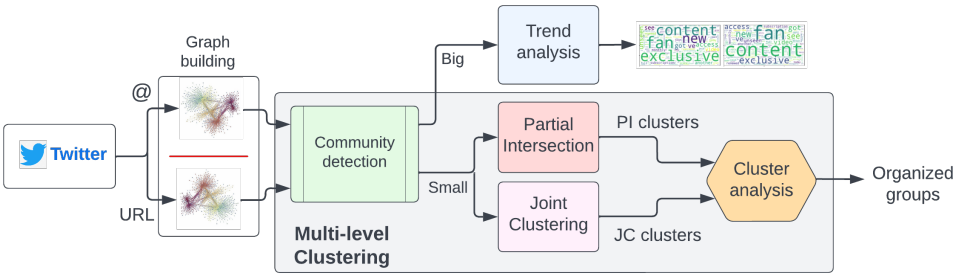


Figure 2.2: Workflow of the proposed Multi-level Clustering

A flowchart showing the proposed method. From the twitter data, the author-URL and author-mention graph are generated following which community detection produces big (>100 nodes) and small (2-99 nodes) communities separately for the two. The big communities are used for trend analysis and the small communities go through two separate grouping processes. Partial intersection finds strict intersecting groups between the two sets of communities and Joint Clustering combines them together to find dense blocks. Both sets of clusters are then analyzed to identify examples of interesting organized groups.

2.4 Methodology

In this section we discuss how we get from collected tweets to organized groups using **MLC**. We introduce relevant notation and delve into details about the proposed method (an overview can be found in Figure 2.2).

2.4.1 Notation

In our dataset, each tweet is posted by an author and contains mentions of other users (Twitter handles) and external URLs (to OnlyFans or other web pages). The author-mention and author-URL information are encoded as non-attributed graphs G_M and G_U respectively with a shared set of author nodes. More specifically,

$$G_M = (\mathbf{N}; \mathbf{E_M});\ G_U = (\mathbf{N}; \mathbf{E_U})$$

$$\mathbf{E_M} = \{(i,j) | n_i.mention \cap n_j.mention \neq \emptyset\}$$

$$\mathbf{E_U} = \{(i,j) | n_i.url \cap n_j.url \neq \emptyset\}$$

where $\mathbf{N} \in \{n_1, n_2, ..., n_{|\mathbf{N}|}\}$ is the set of unique authors in the dataset. Since we conduct our analysis separately for each year, we build a pair of graphs $(G_M, G_U)^{\mathbf{t}}$ for each $\mathbf{t} \in \{2017, 2018, ..., 2021\}$.

For the synthetic experiments and Joint Clustering, which work on attributed graphs, we treat G_M as the adjacency matrix and $\mathbf{X} \in N^{|\mathbf{N}|x|\mathbf{U}|}$ as the attribute matrix created from author-URL information ($\mathbf{U}$ is the set of unique URLs in the data).

2.4.2 Multi-Level Clustering

The proposed method **MLC** is modular and consists of two levels of graph clustering based on i) the network structure and ii) shared attributes. Having obtained clusters from the first level, we further explore two methods of identifying interesting groups. One is called Partial Intersection (PI) which considers strictly intersecting groups of nodes between the clusters and two is Joint Clustering (JC) which implements a second level of clustering on the join of the previous level clusters. As previously discussed, most dense subgraph detection algorithms scale linearly with the graph size and since we are interested in smaller clusters, we opt for this two-pronged approach that first filters out nodes forming large communities based on their network structure.

Community Detection For clustering based on network structure, we rely on community detection to uncover subgraphs with common behaviour (i.e, mentioning common sets of users and URLs). We employ Louvain algorithm (5; 15) which is popularly used on large networks and relies on a two-step process of modularity optimization and community aggregation. The former aims at creating clusters such that their individual modularity is maximized, ensuring that the smallest possible sub-networks of nodes with common behavior are identified. Having found these modular clusters, they are then aggregated to form a new network where the clusters act as new nodes and the whole process is repeated till maximum modularity is achieved.

For each year, we apply Louvain on the corresponding graphs G_M and G_U separately and obtain two sets of communities. Each set is then split into two – communities with

2 to 99 nodes and those with 100 or more nodes (hereby referred to as SMALL and BIG respectively). We suspect that BIG communities carry information about trends on OnlyFans and the overall sex-work industry and we analyse them separately. SMALL communities are more likely to contain smaller organized activities which may potentially be suspicious and remain the main focus of our study.

Input : S_M, S_U, α
Output : IG
IG = $\emptyset$
for c_1 *in* S_M **do**
  for c_2 *in* S_U **do**
    $I = c_1 \cap c_2$
    if $|I| \geq 2$ *AND* $\frac{|I|}{|c_1|} \geq \alpha$ *AND* $\frac{|I|}{|c_2|} \geq \alpha$ **then**
      IG = IG $\cup$ I
    end
  end
end
return IG

Algorithme 1 : This algorithm loops over the communities of authors from the Louvain analysis (S_M and S_U) and returns intersecting groups

Partial Intersection SMALL includes two sets of communities detected from G_M and G_U which we call S_M and S_U. Following this, Algorithm 1 finds the groups that intersect between the two. It loops over both sets of communities and finds groups with at least two common nodes and are 40% of both group sizes or more (i.e, $\alpha = 0.4$). Having obtained a list of intersecting groups, we then filter out groups mentioning fewer than two users and OnlyFans URLs. These groups (called *PI clusters* in Figure 2.2) are then analysed extensively.

Joint Clustering Alternate to strict intersection, we propose an approach that jointly clusters the nodes from SMALL communities to detect latent dense groups. We obtain G_S as the induced subgraph of G_M containing the nodes from both S_M and S_U. G_S acts as the adjacency matrix and $X_S \in N^{|N_S| \times |U_S|}$ is a binary node-attribute matrix where N_S is the set

Input : G_S, X_S, ϕ
Output : Z, *JC clusters*
$Z = G_S\ \phi(X_S)$
JC clusters = HDBSCAN(Z)
return Z, *JC clusters*

Algorithme 2 : This algorithm jointly embeds the node connection (G_S) and content (X_S) information of the induced subgraph and clusters the node embeddings to find latent dense groups

of nodes from G_S and U_S is the set of URLs they mention. X_{ij}=1 if node i mentions URL j and 0 otherwise.

Having obtained G_S and X_S, Algorithm 2 jointly embeds the nodes and their attributes to a low-dimensional Euclidean space $Z \in R^{|N_S| \times d}$ where d is the embedding dimension. These embeddings preserve similarity between nodes in terms of both one-hop neighborhood structure and attribute information. We then detect latent groups by clustering the node embeddings using a hierarchical clustering method like HDBSCAN (6). The clusters thus obtained (called *JC clusters* in Figure 2.2) are further analyzed.

2.5 Experiments and Results

Due to the absence of ground truth in our Twitter data, we verify the effectiveness of **MLC** on synthetic data. In Section 2.5.1, we design a set of synthetic experiments with built-in ground truth to approximate the real-world problem. Next, in Section 2.5.2 we discuss our observations on applying **MLC** to the Twitter dataset and discuss 3 examples of detected organized activity as case studies. This section also contains results from the trend analysis on BIG communities.

2.5.1 Synthetic data experiments

Based on observations on our collected dataset, the real-world graphs are large in terms of number of nodes as well as node attributes and extremely sparse (as indicated by the number of authors, URLs and Mentions in Table 2.1). To approximate this, we generate

Table 2.2: Quality scores (%) comparing the ground truth clusters and predicted clusters on the synthetic graphs. Higher values indicate better similarity between the predicted and true cluster labels.

N (x10^3)	2	6	10	14	18	22	26	30
Louvain (5)	3.9	2.8	2.4	2.3	2.2	2.2	2.1	2.1
Infomap (33)	11.1	11.1	11.1	11.1	11.1	11.1	11.1	11.1
node2vec (14)	3.5	3.5	4.7	4.1	4.5	2.5	2.3	2.5
attri2vec (42)	3.8	4.5	5.7	5.4	7.5	4.2	9.4	8.2
GraphSAGE (18)	3.3	2.2	1.9	1.9	1.9	1.9	2.0	1.75
pcv (29)	5.3	2.4	1.9	1.7	1.6	1.6	1.5	1.5
Partial intersection	4.4	2.4	1.7	1.4	1.2	1.0	0.8	0.7
JC w/KMeans	**11.7**	12.8	13.4	13.2	13.3	13.3	13.4	10.6
JC	9.9	**19.4**	**22.1**	**22.2**	**22.2**	**22.6**	**22.2**	**22.2**

synthetic graphs and inject co-orindated groups that act as ground truth. We then compare the ability of different methods to recover these injected groups.

To generate a synthetic graph $\mathbf{G}=(\mathbf{A};\mathbf{X})$, we assume that both the adjacency matrix $\mathbf{A} \in [0,1]^{n\times n}$ and node attribute matrix $\mathbf{X} \in [0,1]^{n\times m}$ of the graph are modelled by Bernoullli random variables whose distribution can be specified through the latent groups. We set values for $\mathbf{P} \in [0,1]^{g\times g}$ and $\mathbf{Q} \in [0,1]^{g\times m}$ such that $\mathbf{P_{ii'}}$ is the probability of an edge existing between two nodes belonging to groups i and i', and $\mathbf{Q_{ij}}$ indicates the probability of a user from latent group i having an attribute from latent group j. We also set $\mathbf{Z_s} \in \{0,1\}^{nxg}$ as a binary matrix that encodes hard partition of n nodes into g non-overlapping latent node groups. $\mathbf{Z_s}$ is analogous to $\mathbf{Z}$ from Algorithm 2 in the sense that it would contain dense regions representing the injected groups. Specific details about the values set for $\mathbf{P}, \mathbf{Q}$ and $\mathbf{Z}$ can be found in (41).

We inject eight coordinated groups (so g=9 corresponding to 8 injected latent coordinated groups and 1 group of normal users) with 20 nodes and 20 attributes on differently sized graphs (from 2,000 to 30,000 nodes). This means that the ratio of coordinated to normal nodes gradually decreases from 8% to 0.5% of the graph size, making it increasingly more challenging to detect them.

We run the following baselines on our synthetic data: Infomap (33) and Louvain (5) from community detection; pcv (29) from dense subgraph or tiny cluster detection; node2vec (14), attri2vec (42) and unsupervised GraphSAGE (18) from network embeddings. Pcv only considers node contents, Infomap, Louvain and node2vec only consider connections, and the other baselines incorporate both content and connections. We compare two versions of the Joint Clustering algorithm where the clustering is done using KMeans with k=9 (JC w/KMeans) and HDBSCAN (original JC). Lastly we also compare the Joint Clustering results with a Partial intersection approach. To evaluate partitions (how well coordinated groups are separated from the background and each other), we use the quality score in (29) which rewards the cluster label alignment that maximizes Jaccard's similarity between the predicted and ground truth cluster labels.

We repeat each experiment 4 times and the average quality scores are reported in Table 2.2. JC methods outperform all baseline methods and especially improves with increase in the number of users where injected groups occupy only a small fraction (0.5%). JC (with HDBSCAN) performs significantly better than JC w/KMeans and Partial intersection. However, it is to be noted that for a fair comparison with JC, the graphs for PI were setup differently for the synthetic experiments than in **MLC**. The former treats G_M as the adjacency matrix and builds a node attribute matrix using G_U whereas the latter considers them as two separate graphs. Nonetheless, it is an important baseline because it also acts as an ablation study where we compare the strict intersection of the Louvain communities with the clusters detected by jointly considering connections and content.

We also examine the node embeddings learned by JC and compare the separability of the injected organized groups from the normal nodes for multiple baselines. Figure 2.3 shows that JC does a better job compared to other embedding based methods.

2.5.2 Real world data

Text clustering experiments Before applying **MLC** on our real-world Twitter data, we attempted to understand this data by simply clustering the tweet texts. To this end, we

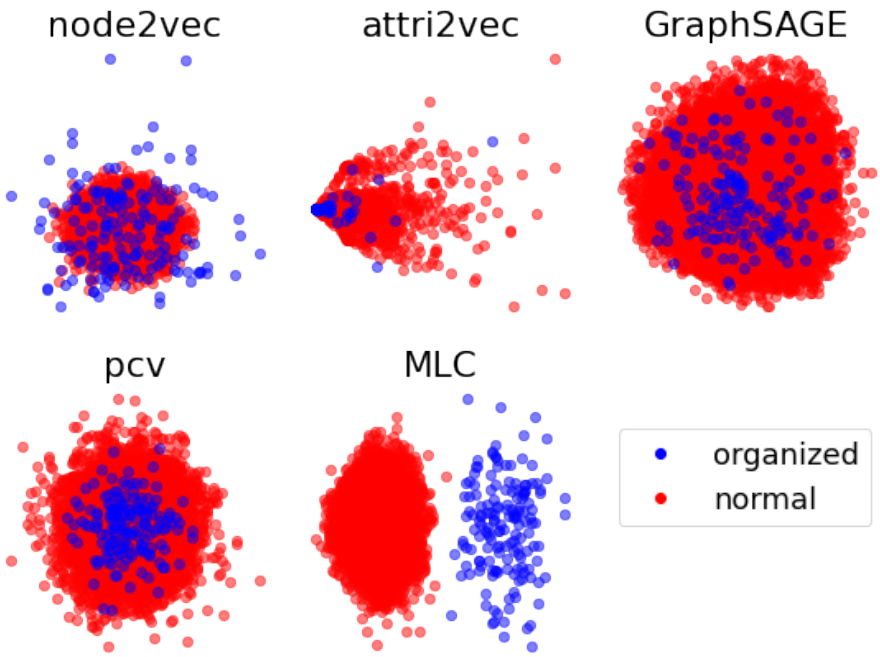


Figure 2.3: The embeddings learned by JC projected on lower dimensions (using PCA) shows clean separation between normal data points and those belonging to injected organized groups.

applied KMeans (with K=5) separately on TFIDF representations of 250,000 tweets from years 2017 and 2021. We observed that both years contained a predominant cluster that made up more than 90% of the total tweets. To better examine the clusters, we looked into the tweets closer to each cluster's centroid and concluded that most clusters contained the same theme: the promotion of one or more OnlyFans accounts using generic sexually explicit language. Most of these messages seemed to have been automatically generated from the OnlyFans platform. What seemed to differentiate most of these clusters were the structure of their text (templates). We further applied KMeans on the biggest cluster (90% of tweets) using K=5. Once again, the tweets closer to the centroid of each cluster accounted for the generic promotion of OnlyFans accounts. Almost all clusters contained the expression "unseen and exclusive content" with URLs attached to their tweets. Thus, we concluded that simple text clustering methods can only give us limited insights.

We also applied a text clustering and template detection algorithm specifically de-

signed for HT detecting from escort ads called Infoshield (24). Similar to KMeans, the results were not informative and predominantly contained automatic templates from OnlyFans. Although Infoshield works well on escort ad text, this method is not as helpful in short tweet texts as seen in our data. These experiments tell us that we can not rely solely on textual information for detecting organized activity. This further bolsters our approach of focusing on user-interaction activity, i.e. user mentions and linked URLs.

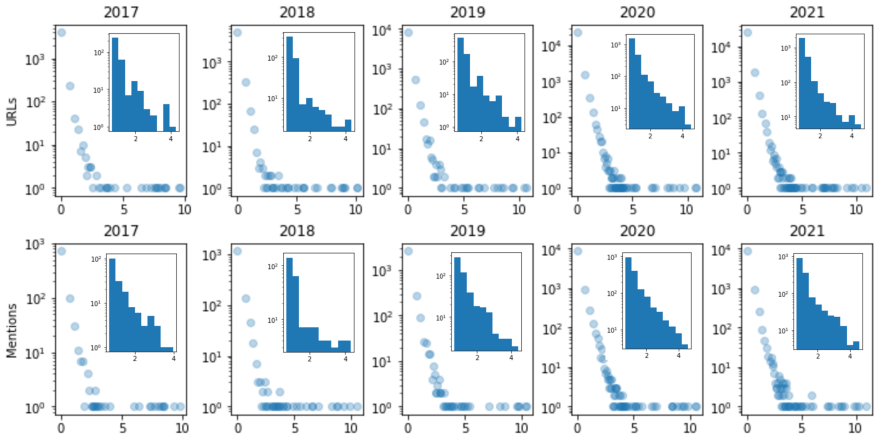


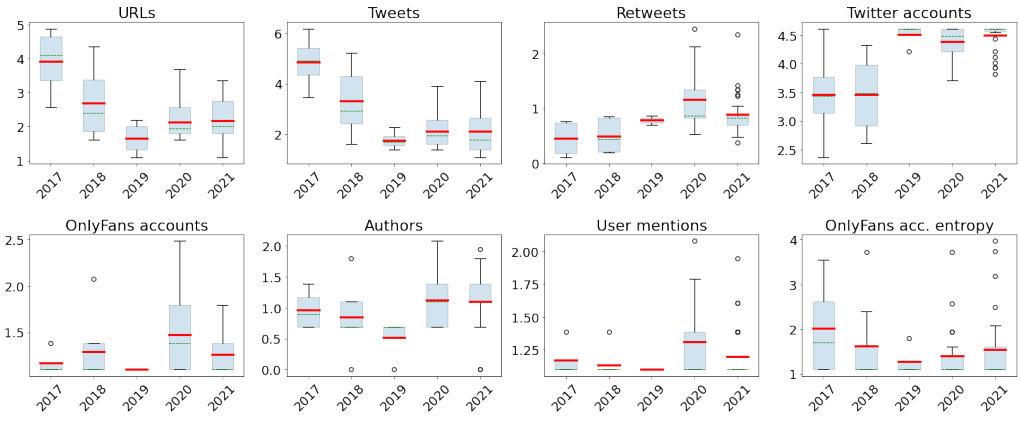
Figure 2.4: Community size distribution for URL graph and Mention graph for every year. The insets are zoomed in distributions for the SMALL communities. Plots are in log-log scale.

community$_d istributions$

Table 2.3: Year-wise clusters statistics detected from the SMALL and BIG communities.

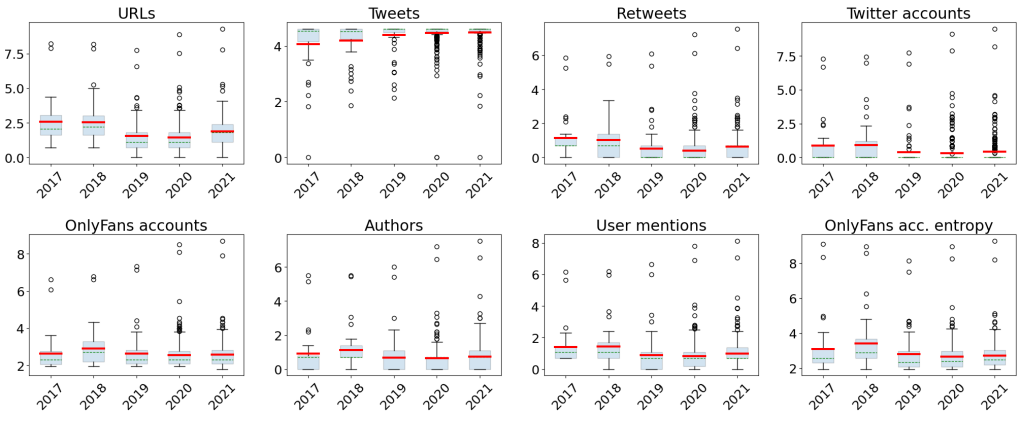
	Partial Intersection (SMALL)				Joint Clustering (SMALL)		Community Detection (BIG)	
Year	Mentions	URLs	Inter.	Filtered	Clusters	Filtered	Mentions	URLs
2017	177	346	91	4	36	23	12	14
2018	245	452	111	8	43	25	13	15
2019	492	781	204	5	80	18	16	14
2020	1,679	2,263	660	21	258	81	14	19
2021	1,498	2,680	695	34	235	103	20	21

Community detection We uncover author communities through their shared Mentions and URLs separately (S_M and S_U) for each year and study the distribution of their sizes in



[b]

Figure 2.5: Partial Intersection



[b]

Figure 2.6: Joint Clustering

Figure 2.7: The properties of the detected organized groups over the years. The y-axis indicates counts in log scale. The red line shows the mean and the green dotted line is the median. Retweets are in percentage and everything else is a count.

Figure 2.4. We observe that this distribution resembles a power law where majority of the communities consist of unique nodes or lone authors. We discard these nodes from the rest of our analysis as they are not truly "communities". The inset histograms zooming in on the SMALL communities show that in most cases, their size distribution decreases gradually.

The total number of communities per year for both SMALL and BIG is mentioned in the first two columns of Table 2.3. Within SMALL communities we observe that for every year, Mentions contain fewer communities than URLs. URL communities are 125% more than Mention communities in 2017 and 180% more in 2021. This is not surprising considering that the total number of URLs in our collected data is substantially higher than the total number of Mentions (see Table 2.1). On the other hand, since there are much fewer BIG communities, their numbers stay relatively constant through the years.

Partial Intersection After applying Algorithm 1 on SMALL communities, we find intersecting groups of Mentions and URLs as shown in column 'Inter.' in Table 2.3. Note that the column 'Filtered' refers to the number of intersecting groups containing multiple user mentions and OnlyFans accounts. We also present the boxplots of the number of tweets, authors, user mentions, OnlyFans accounts, URLs and Twitter accounts found in the PI clusters in Figure 2.5. We include further discussion on these results in Section 2.6.

Joint Clustering For JC, we use Singular Vector Decomposition as ϕ to obtain Z, projecting X_S to its first d left singular vectors with $d = 100$. The number of joint clusters found using HDBSCAN for each year are mentioned in the 'Joint Clustering (SMALL)' column of Table 2.3. Although the number of JC clusters found each year is lower than that of PI clusters, JC produces more 'Filtered' clusters containing multiple user mentions and OnlyFans accounts than PI. This supports our hypothesis that joint clustering using both Mentions and URLs is better at finding organized groups. Figure 2.6 shows the boxplot of the cluster properties. We also discuss some interesting cluster examples in Section 2.6.

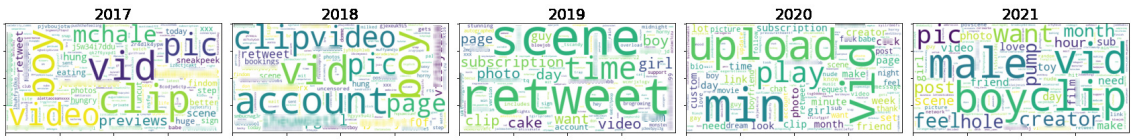


Figure 2.8: Word Cloud of the most popular and topical words from BIG communities for each year. The size of a word is indicative of its frequency. Words referring to specific usernames or explicit in nature have been blurred.

Trend analysis We conduct an experiment on the BIG communities to better understand the trends in larger groups in our data across the years. For a particular year, we apply separately, a Latent Dirichlet Allocation (LDA) model (4) on the tweets of each cluster, which gives us latent topics present in the text. Although topic modeling is not very helpful due to the extremely short nature of the text, we group together the top 5 words from the top 10 topics for each community within a year. This provides us with a list of highly popular and topical words representing the BIG communities in a year and their word cloud is presented in Figure 2.8. Similar to our analysis using text clustering (in paragraph 2.5.2), we do not observe a drastic change in the top words from 2017 to 2021. Most of the words including *'video', 'clip', 'account', 'retweet', 'subscription'*, point towards the promotion of OnlyFans content. However, we do notice the presence of more words such as *'boy'* and *'male'*. Considering that most of these tweets are self-promotional or promoting other accounts, this could possibly hint towards a bigger than expected male-to-female ratio of OnlyFans creators in our data sample.

2.6 Case studies and Discussion

To identify interesting examples from the set of organized groups detected by **MLC** (from both Filtered columns in Table 2.3), we jointly consider the number of tweets, the number of OnlyFans accounts promoted and the frequency of their promotion within a cluster. For e.g, consider a cluster with 100 tweets promoting 2 OnlyFans accounts A and B. A and B being promoted 2 and 98 times respectively is less interesting than 49 and 51 times. In other words, it is more interesting or suspicious if A and B are almost always promoted

together. We capture this metric in the 'OnlyFans acc. entropy' boxplot in Figure 2.7. We find three interesting cases and discuss them below.

Case study 1 Our first case involves the coordinated promotion of two OnlyFans accounts found in two *PI clusters*, one each from 2017 and 2018. For the sake of privacy and clarity, we name these accounts CS_{1A} and CS_{1B}. On closer inspection, we find that 75% (of 479 tweets) from the group in 2017 and 63% (of 57 tweets) from the group in 2018 were promoted by the account owners (OnlyFans creators) themselves. Consequently, a majority of these tweets were automated messages from OnlyFans, this is also indicated by their API source available in the data. 32 authors in 2017 and 8 authors in 2018 (other than the owners) promoted CS_{1A} and CS_{1B}. Three of these authors appeared both in 2017 and 2018, which further confirms the coordination between both accounts. Each account is promoted separately over a 100 times in 2017 and 10 times in 2018 and together in 12 tweets in 2017 and 1 tweet in 2018. As of today, the OnlyFans account of CS_{1A} no longer exists, but its twitter account is still active. CS_{1A} describes themself as a photographer and claims to be married to CS_{1B}. This coordinated promotion also appears in the *JC clusters* from 2017 and 2018. Additionally, we also find another account (additional to CS_{1A} and CS_{1B}) that reappears in both years. From these observations, we note that these two accounts are coordinated but show no further signs of suspicious activity.

Case study 2 In 2020 we found a *PI cluster* of 40 authors promoting seven OnlyFans accounts through 48 retweets. Most of these OnlyFans accounts were promoted simultaneously and 40% of the tweets in the cluster promoted at least two accounts. From this, it can be inferred that these accounts are aware of one another. *JC clusters* from 2020 also contained a small group of 15 tweets mentioning 5 out of these 7 accounts. What stands out here is that the ratio of tweets to authors in this cluster is close to 1 and all the 5 accounts were promoted by a different set of 14 authors. Again there is no indication of HT, but we clearly see that there is coordinated behaviour.

Case study 3 In 2020, we found a *JC cluster* containing 25 tweets promoting 8 OnlyFans accounts. All 8 accounts were promoted by a separate group of 7 authors with Twitter handles that follow a specific pattern starting with the word 'OnlyFans' followed by promotional words such as 'boost' or 'promo'. It appears that these Twitter accounts were created to promote OnlyFans accounts and their tweets often include #OnlyFansPromo. According to their profiles, they promote tweets or provide retweets for a fee starting at $10. This shows that **MLC** is able to establish connections between similar third parties promoting the same accounts. However, the authors of these tweets are payed promoters, in other words this cluster do not show signs of suspicious activity.

2.7 Discussion

In this section, we discuss the main conclusions of our study and summarize them in the following points.

Growing number of communities over the years From Table 2.3, we see an increase in the number of communities between 2017 and 2020 (numbers stay stable between 2020 and 2021). Apart from 2017, this can not be attributed to the total number of collected tweets (Table 2.1) but to the steady increase in the number of authors, URLs and Mentions. Additionally, it can also be explained by the Twitter trends related to OnlyFans promotions with sharp jumps from 2018 to 2019 and 2019 to 2020 as shown in Figure 2.1. The number of *PI clusters* and *Joint clusters* also follows the same trend as they are built off of the communities. The boxplots in Figure 2.7 show the differences in the two sets of detected clusters. The average values across each graph within Partial Intersection (Subfigure 2.5) varies with the year with high variances compared to Joint Clustering. Year 2019 stands out with regards to URL, OnlyFans and tweets counts, but shows a higher average of Twitter accounts than the other years. This explains why 2019 had fewer interesting cases to investigate. On the other hand, Joint clusters tend to have steady yearly averages and variances in each of the properties. Furthermore, *JC clusters* contain

many more outliers and are more homogeneous than *PI clusters*.

Presence of coordinated activities in tweets promoting OnlyFans Previous works (39) have concluded that OnlyFans creators often engage in social instrumental support for various reasons, including promotion of their content. One such collaboration is 'Share-for-Share' where a creator shares the posts of another creator to get their own posts shared by them in return. Some creators also rely on paid promotions which generally involve a third party charging a specific fee per post/share.

Another type of activity involves fake profiles where users may be tempted to create fake profiles, either automated or manual, to gain more followers and promote themselves (16). Such accounts are termed as "sock-puppets" and have been studied in the literature (23). This can lead to a scenario where an agent inadvertently creates a seemingly coordinated network, when in fact this network is controlled by the same person. This does not necessarily mean that the tweets in question are not suspicious. The agent could either be the creator themself or a trafficker controlling the creator account. Moreover, in social media, agents can appear once and disappear forever, or reappear using a new account.

In short, there are multiple scenarios to consider when it comes to classifying coordinated networks as suspicious or not. The situations discussed above are often aimed at supporting smaller creators and the networks thus formed can get detected by algorithms such as ours, mistaking them for being nefarious networks and classifying them as suspicious. Hence it is imperative to try and understand the social network under study and be responsible while assigning labels. Based on this, we conclude that the organized activity results presented in Table 2.3 are not meant to be seen as hard proof of coordinated organized crime groups, but rather as potential preliminary leads for investigation.

Conclusions from case studies Our analysis of the case studies combined with the above concepts help us outline a conclusion for each case. Case study 1 is most likely a husband-wife duo promoting each other (as mentioned in the husband's account de-

scription). We speculate that the additional authors found in the cluster may be part of a 'Share-for-Share' network, paid promoters or manually created fake accounts. Case study 2 shows some signs of a coordinated network, where a group of authors were promoting a different set of accounts seemingly belonging to men of a specific ethnic group. This may be a group of young men trying to help each other by sharing their connections. However, based on the information we have, even though men are usually less prone to be targets of HT than women, we can not clearly conclude that this group shows no signs of suspicious behaviour. Case study 3 can be easily declared as a case of paid promotion where all the posts were promoted by a paid group of Twitter accounts specifically focused on OnlyFans promotions.

Case building and detecting clues As explained previously, to help in our investigation, we filtered out clusters based on the number of mentions and OnlyFans accounts. These filters are based on the idea that HT being an organized activity involves multiple victims (24) and therefore will advertise multiple accounts. We observe that groups with very few tweets are naturally less informative, and apart from the number of OnlyFans accounts advertised, we also need to consider the frequency of their promotions. For e.g, if a group of authors promotes an OnlyFans account only once while promoting another account on 50 out of 51 tweets, it is difficult to conclude that these two accounts are part of the same network. Thus, the groups detected by **MLC** need to be carefully analyzed and filtered to aid with case building.

Advantages of a two-pronged clustering approach Although the Joint clustering approach performs better than Partial intersection in the synthetic experiments, we suspect this may be partially attributed to the nature of how the graphs were setup in the experiment and observe that it is still able to retrieve interesting example cases. Case studies 1 and 2 were identified from *PI clusters* whereas JC partially corroborated them. Although the PI approach gave us a wider pool of interesting (higher OnlyFans entropy for example) than JC clusters, JC did provide Case study 3. Hence, we believe that **MLC** using both

clustering methods in tandem makes best usage of both the explicit network structure and the embedded attribute information, leading to a more thorough detection of organized groups.

Trends of OnlyFans promotion on Twitter This study is the first to mine and analyse millions of tweets related to OnlyFans. Figure 2.1 shows that the growing number of tweets are directly connected to the surge of OnlyFans in the last few years, and it is most likely that this number will continue to flourish through time, inevitably creating more and more coordinated networks. Simple text analyses of our data show that a large number of these tweets are generic and focus on promoting OnlyFans accounts. Moreover, we observe the sexually explicit vocabulary present in the tweets remain constant through time. Finally, we also notice certain words such as 'boy' and 'male' showing up in the tweets, which may hint towards higher than expected male to female creator ratio. This is also important to note because currently most studies on HT detection or OnlyFans creators are focused only on females.

Scope for further research This paper demonstrates that community detection and dense block detection are well-suited for finding organized groups of OnlyFans advertisements on Twitter. Specifically for measuring "suspiciousness", there are certain clues and/or metrics based on which detected groups can be ranked. Although we currently do this in a way by looking at the OnlyFans account entropy of clusters, this can be more systematically formalized. Lastly, considering that ours is a first exploratory study on this data, we only considered user mentions (as connectivity) and URLs (as content), future works may incorporate additional matrices create from features such as: hashtags, keywords in the tweets, OnlyFans accounts and more. OnlyFans accounts is specially interesting since this last metric was one of our main point of interest when measuring the quality of a cluster.

References

[1] Adhoob, M. (2021). Trafficking in persons. *IELR*, 37:273.

[2] Akhund, T., Pollack, D., and Shipp, K. (2022). Human trafficking and technology.

[3] Alvari, H., Shakarian, P., and Snyder, J. (2017). Semi-supervised learning for detecting human trafficking. *Security Informatics*, 6(1):1–14.

[4] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[5] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

[6] Campello, R. J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

[7] Chavoshi, N., Hamooni, H., and Mueen, A. (2016). Debot: Twitter bot detection via warped correlation. In *Icdm*, pages 817–822.

[8] Comerford, T. (2022). Pornography isn't the problem: A feminist theoretical perspective on the war against pornhub. *BCL Rev.*, 63:1177.

[9] Deshpande, Y., Sen, S., Montanari, A., and Mossel, E. (2018). Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31.

[10] Fortunato, S. and Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41.

[11] Giammarinaro, M. G. (2020). Covid-19 position paper: The impact and consequences of the covid-19 pandemic on trafficked and exploited persons. *Retrieved from COVID-19-Impact-trafficking. pdf (ohchr. org)*.

[12] Giatsoglou, M., Chatzakou, D., Shah, N., Beutel, A., Faloutsos, C., and Vakali, A. (2015a). Nd-sync: Detecting synchronized fraud activities. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 201–214. Springer.

[13] Giatsoglou, M., Chatzakou, D., Shah, N., Faloutsos, C., and Vakali, A. (2015b). Retweeting activity on twitter: Signs of deception. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 122–134. Springer.

[14] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

[15] Guillaume, L. (2008). Fast unfolding of communities in large networks. *Journal Statistical Mechanics: Theory and Experiment*, 10:P1008.

[16] Gurajala, S., White, J. S., Hudson, B., Voter, B. R., and Matthews, J. N. (2016). Profile characteristics of fake twitter accounts. *Big Data & Society*, 3(2):2053951716674236.

[17] Hamilton, V., Soneji, A., McDonald, A., and Redmiles, E. (2022). " nudes? shouldn't i charge for these?": Exploring what motivates content creation on onlyfans. *arXiv preprint arXiv:2205.10425*.

[18] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[19] Hernández-Álvarez, M. (2019). Detection of possible human trafficking in twitter. In *2019 International Conference on Information Systems and Software Technologies (ICI2ST)*, pages 187–191. IEEE.

[20] Hooi, B., Song, H. A., Beutel, A., Shah, N., Shin, K., and Faloutsos, C. (2016). Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 895–904.

[21] Jiang, M., Cui, P., Beutel, A., Faloutsos, C., and Yang, S. (2016). Inferring lock-step behavior from connectivity pattern in large graphs. *Knowledge and Information Systems*, 48(2):399–428.

[22] Kulshrestha, A. (2021). *Detection of Organized Activity in misc Escort Advertisements*. McGill University (Canada).

[23] Kumar, S., Cheng, J., Leskovec, J., and Subrahmanian, V. (2017). An army of me: Sockpuppets in misc discussion communities. In *Proceedings of the 26th International Conference on World Wide Web*, pages 857–866.

[24] Lee, M.-C., Vajiac, C., Kulshrestha, A., Levy, S., Park, N., Jones, C., Rabbany, R., and Faloutsos, C. (2021). Infoshield: generalizable information-theoretic human-trafficking detection. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1116–1127. IEEE.

[25] Li, L., Simek, O., Lai, A., Daggett, M., Dagli, C. K., and Jones, C. (2018). Detection and characterization of human trafficking networks using unsupervised scalable text template matching. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3111–3120. IEEE.

[26] López, C. (2020). A wave of people turned to onlyfans to earn money when they lost their jobs due to the pandemic. *Insider. https://www. insider. com/people-are-creating-onlyfans-accounts-after-losing-jobs-during-pandemic-2020-6*.

[27] Mensikova, A. and Mattmann, C. A. (2018). Ensemble sentiment analysis to identify human trafficking in web data. In *Workshop on Graph Techniques for Adversarial Activity Analytics (GTA 2018), Marina Del Rey, CA, USA*, pages 5–9.

[28] MINOR, S. D. (2015). A report on the use of technology to recruit, groom and sell domestic minor sex trafficking victims.

[29] Neumann, S. (2018). Bipartite stochastic block models with tiny clusters. *Advances in Neural Information Processing Systems*, 31.

[30] Polaris (2021). 2020 us national human trafficking hotline statistics. https://polarisproject.org/2020-us-national-human-trafficking-hotline-statistics/.

[31] Rabbany, R., Bayani, D., and Dubrawski, A. (2018). Active search of connections for case building and combating human trafficking. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2120–2129.

[32] Ramchandani, P., Bastani, H., and Wyatt, E. (2021). Unmasking human trafficking risk in commercial sex supply chains with machine learning. *Available at SSRN 3866259*.

[33] Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4):1118–1123.

[34] Shin, K., Eliassi-Rad, T., and Faloutsos, C. (2016a). Corescope: Graph mining using k-core analysis—patterns, anomalies and algorithms. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 469–478. IEEE.

[35] Shin, K., Hooi, B., and Faloutsos, C. (2016b). M-zoom: Fast dense-block detection in tensors with quality guarantees. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 264–280. Springer.

[36] Shin, K., Hooi, B., Kim, J., and Faloutsos, C. (2017a). D-cube: Dense-block detection in terabyte-scale tensors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 681–689.

[37] Shin, K., Hooi, B., Kim, J., and Faloutsos, C. (2017b). Densealert: Incremental dense-subtensor detection in tensor streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1057–1066.

[38] Tong, E., Zadeh, A., Jones, C., and Morency, L.-P. (2017). Combating human trafficking with deep multimodal models. *arXiv preprint arXiv:1705.02735*.

[39] Uttarapong, J., Bonifacio, R., Jereza, R., and Wohn, D. Y. (2022). Social support in digital patronage: Onlyfans adult content creators as an misc community. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–7.

[40] Vlerick, M. and Van Hove, J. (2021). Revolutionising digital sex work: an analysis of the impact of onlyfans on sex workers.

[41] Wang, J., Levy, S., Wang, R., Kulshrestha, A., and Rabbany, R. (2019). Scg: Spotting coordinated groups in social media. *arXiv preprint arXiv:1910.07130*.

[42] Zhang, D., Yin, J., Zhu, X., and Zhang, C. (2019). Attributed network embedding via subspace discovery. *Data Mining and Knowledge Discovery*, 33(6):1953–1980.

[43] Zhang, S., Zhou, D., Yildirim, M. Y., Alcorn, S., He, J., Davulcu, H., and Tong, H. (2017). Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 570–578. SIAM.

General Conclusion

Online human trafficking has increased over the last few years. Furthermore, Covid-19 has isolated vulnerable people while preventing help interventions. Meanwhile, social media has taken over the world. This amalgam of conditions has caused platforms such as OnlyFans to thrive. OnlyFans has been advertised as an easy and safe way to make money, which has been particularly appealing to people going trough financial hardships. However, the platform doesn't have security measures to prevent illegal content to be posted. These conditions lead us to believe that the platform allows, inadvertently, the presence of human trafficking. In this thesis we propose a simple method to help investigators find a way to narrow down their search. Since OnlyFans accounts are only accessible via a paywall we propose an unsupervised lockstep behaviour detection method that investigates public Tweets advertising the OnlyFans accounts. Lockstep detection as well as community detection are popular social media analysis methods to study the activities between users. Before arriving to this solution however, based on prior work on HT that heavily relied on NLP methods, we decided to start our research by understanding the data and testing potentials of NLP method.

Understanding the data We started our analysis using a collection of 16 million tweets mentioning the word OnlyFans between 2020 and 2021. These tweets also contained the tweet id, author id, indication of if it is an original tweet or Retweet and geolocation. Please note that we used an improved, separate data collection for the main study in Chapter 2, which is explained later on and collects more meta data.

One of the first analysis done on this prior data set was to perform keyword analy-

sis. We first verified that the tweets were in fact promoting sex work content associated with OnlyFans, our results gave us clear indications that the vocabulary present in our corpus was related to sex work and the promotion of it. Furthermore, we noticed that there were subtle changes in the vocabulary between the period 2017-2018 versus period 2019-2020. In fact, the latter period contained more emoticons. Putting aside the emoticons, the vocabulary was overall similar. In our paper, we did a similar analysis but instead of investigating all the data for every year, we investigated the tweets within communities containing 100 nodes and more (see Section 2.5.2). We also noticed that the vocabulary didn't change much, but ,the usernames present in the corpus shows that the ratio male to female is higher than expected. The other interesting experience we ran on our prior tweet collection is the simple topic analysis using KMEANS (with k=5) on TF-IDF tweet representations. We based our analysis on sample data containing 250K tweets chosen randomly per year. We first investigated the evolution of our clusters (derived from KMEANS) through time by basing our analysis on year 2017, then on year 2021. Both 2017 and 2021 show a cluster containing more than 90% of all tweets, this cluster became increasingly bigger through the years for both analysis. Further analysis of the clusters,as explained in our paper (see Section 2.5.2), shows that the 'theme' was the same for all our tweets: promotion of a or various OnlyFans accounts. In short, What differentiated a cluster of tweet to another was their specific template.

Community detection analysis and Infoshield Next, we conducted a community detection experiment on our data, in addition to applying the Infoshield clustering (which clusters the tweets coming from the same template). After extracting specific features from our data, we realized that Username Mentions and URLs, due to their volume, would be the most interesting to study. Note how we carried this specific feature analysis on to the new data collection (our paper). Our method is divided in two steps, the first step is made of two approaches, we first find the (Louvain) communities based on URLs and Mentions (derived from graphs *GU* and *GM* just as in Section 2.4.2 from our paper), in the second approach we apply Infoshield on the every tweet (text) and find clusters with

similar templates. In the second step we simply find the partial intersections between these three clusters (using the same algorithm as the one in the paper see algorithm 1. In other words, our methodology simply finds all the groups of authors that are simultaneously sharing the same URL links, Username Mentions and templates. For 2017, the only year we performed this analysis, the number of intersections between Mentions and URLs communities is 44. This is substantially less than what we found with the dataset used in the paper (91 intersection in 2017), this can easily be explained by two factors. In fact, in the data collection used for this analysis, the URLs and Mentions were extracted and were not taken from the API like we did in the paper, furthermore, in this analysis the Small communities are comprised of 3 to 99 nodes instead of 2 to 99 nodes- we discarded communities containing 2 nodes and by doing so, we removed a substantial amount of communities. Regardless of the differences, what is of interest in this analysis is the intersection between these communities and Infoshield. We found 16 intersections; we lost 28 clusters by applying Infoshield. Meaning that it kept only the intersections containing authors promoting the same templates. This, instead of helping with our research, potentially removed clusters containing interesting interactions. We therefore decided that Infoshield should be removed from our pipeline.

Summary and Additional points from the discussion in Section 2.7 In our discussion, we explained the growing number of communities through the years (see table 2.3). We noted that, this phenomenon can not be explained by the yearly number of tweets we collected but can be explained by the increasing number of URLs and Mentions (yearly - see table 2.1) of our dataset. This can be in turn explained by figure 2.1 where we see that the total number of Tweets (along with the total number of authors) containing the word OnlyFans grows yearly. These increases are particularly notable between years 2018 to 2020. Another point we highlighted in the discussion, is the difference between clusters derived from our dense-block method (Joint Clustering approach) versus the clusters derived from our Partial Intersection approach. In short, our JT clusters composition is different from the data we find in our Partial Intersections. This difference is carried

when we filter for the clusters (JT clusters and PI) containing multiple Mentions and OnlyFans accounts. This difference also has an important impact in the performance when running the synthetic experiment. We shortly explain this issue in the paper (see Section 2.4.1 and Section 2.5.1), the following paragraph offers a complete explanation. If we recall, JT clustering outperforms every other baseline, including Louvain. However, Partial Intersection performs worst than Louvain alone. This is surprising since in the case-by-case analysis (after filtering) PI clusters provided more interesting clusters than PJ clustering. In fact, overall, the filtered PI are of higher quality than JT clusters – they contain a higher OnlyFans account entropy, meaning that when a PI cluster contains multiple OnlyFans accounts, each one of these accounts is promoted in closer proportion to each other (See Section 2.6 for more explanation on entropy). In fact, when deciding which cases would be presented in the paper, we had far more choices among the PI clusters. In short, PI results in table 2.2, are in fact not representative. The major reason for this discrepancy is that in our synthetic experiments and in our Joint Clustering method, graphs author-mention and author-URL are represented as the adjacency matrix and node attribute matrix of a graph. This is different from the setup in Partial Intersection where we work on two adjacency matrices alone and do not consider one of them as node attributes. Since the data representation in the synthetic experiments differs from the PI method, we can therefore discard the performance output and draw our conclusion solely based on the cases we found. On the other hand, considering the case-by-case analysis of clusters, JT clustering seem to have under-performed compared to our PI approach. Overall however, our **MLC** is able to find potential coordinated behaviour between users, but the raw numbers derived from table 2.3 can't be classified as suspicious coordinated clusters. However, further investigation can help determine if in fact a group is coordinated and if we want to establish whether or not the group is suspicious, we need to follow specific cues. In fact, various scenarios can arise that might appear suspicious but are not. Among other strategies, creators inadvertently create coordinated networks by paying for promotion or by participating in the share-per-share practices, by digging into the data we are able to see this type of coordination and confirm that a cluster is not suspicious.

For example, Case number 3 (see Section 2.6) presented in our paper was classified as non suspicious since we know that all the authors of these tweets were payed. To conclude on this point, we can say that our method **MLC** is effective in finding coordinated behaviour by filtering our PI and JT clusters and by investigating further these filtered clusters (analysing Entropy and looking for clues). As mentioned in the paper, these are steps/metrics that can be systematically formalized in the future. When it comes to future research, we must reiterate that additional feature analysis can be beneficial to our model. We can even go further by stating that we should discard URL analysis in favor of OnlyFans analysis and possibly paying more attention to Twitter accounts. In fact, unique OnlyFans accounts as well as unique Twitter accounts were extracted from the URLs after creating the graphs GM and GU. In short, we propose the creation of Graphs GO and GT, this will necessitate more preliminary work, but will potentially achieve better results. On the other hand, we could run the method on Tweets containing other keywords; sex work is now always evolving in the web giving the opportunity to traffickers to promote new platforms or new trends on Twitter. Our last observation is that, we could emulate paper (1) by taking full advantage of the URL feature and extracting the pictures attached to them, to then run a computer vision method on those images, thus adding to the information we found (example the sex of the individual).

Bibliography

[1] Adhoob, M. (2021). Trafficking in persons. *IELR*, 37:273.

[2] Akdogan, A. (2021). Word embedding techniques: Word2vec and tf-idf explained.

[3] Akhund, T., Pollack, D., and Shipp, K. (2022). Human trafficking and technology.

[4] Alvari, H., Shakarian, P., and Snyder, J. (2017). Semi-supervised learning for detecting human trafficking. *Security Informatics*, 6(1):1–14.

[5] Anthony, B. (2018). On-ramps, intersections, and exit routes: A roadmap for systems and industries to prevent and disrupt human trafficking.

[6] Aslam, S. (2022). Twitter by the numbers: Stats, demographics fun facts.

[7] Barzilay, R. and Lee, L. (2003). Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. *arXiv preprint cs/0304006*.

[8] Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

[9] Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008.

[10] Bouché, V. (2015). A report on the use of technology to recruit, groom and sell domestic minor sex trafficking victims.

[11] Boyagane, I. (2020). How important the words in your text data? tf-idf answers....

[12] Campbell, S. (2022). Onlyfans statistics 2022: How many people use onlyfans?

[13] Campello, R. J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

[14] Chavoshi, N., Hamooni, H., and Mueen, A. (2016). Debot: Twitter bot detection via warped correlation. In *Icdm*, pages 817–822.

[15] Christensen, A. P., Garrido, L. E., and Golino, H. (2020). Comparing community detection algorithms in psychological data: A monte carlo simulation.

[16] Comerford, T. (2022). Pornography isn't the problem: A feminist theoretical perspective on the war against pornhub. *BCL Rev.*, 63:1177.

[17] D. Zhang, Y. Jie, X. Z. and Zhang, C. (2020). Node2vec explained.

[18] Dabbura, I. (2018). K-means clustering: Algorithm, applications, evaluation methods, and drawbacks.

[19] Dangeti, P. (2023). Statistics for machine learning: The elbow method.

[20] Deshpande, Y., Sen, S., Montanari, A., and Mossel, E. (2018). Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31.

[21] do Prado, K. S. (2017). How dbscan works and why should we use it?

[22] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *kdd*, volume 96, pages 226–231.

[23] fandafia (2022). Promoting onlyfans on twitter: 11 powerful methods that work.

[24] Fansly and Creators, O. (2022). How to promote the onlyfans page on twitter?

[25] Fortunato, S. and Barthelemy, M. (2007). Resolution limit in community detection. *Proceedings of the national academy of sciences*, 104(1):36–41.

[26] Giammarinaro, M. G. (2020). Covid-19 position paper: The impact and consequences of the covid-19 pandemic on trafficked and exploited persons. *Retrieved from COVID-19-Impact-trafficking. pdf (ohchr. org)*.

[27] Giatsoglou, M., Chatzakou, D., Shah, N., Beutel, A., Faloutsos, C., and Vakali, A. (2015a). Nd-sync: Detecting synchronized fraud activities. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 201–214. Springer.

[28] Giatsoglou, M., Chatzakou, D., Shah, N., Faloutsos, C., and Vakali, A. (2015b). Retweeting activity on twitter: Signs of deception. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 122–134. Springer.

[Github] Github.

[30] Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

[31] Guillaume, L. (2008). Fast unfolding of communities in large networks. *Journal Statistical Mechanics: Theory and Experiment*, 10:P1008.

[32] Gurajala, S., White, J. S., Hudson, B., Voter, B. R., and Matthews, J. N. (2016). Profile characteristics of fake twitter accounts. *Big Data & Society*, 3(2):2053951716674236.

[33] Hamilton, V., Soneji, A., McDonald, A., and Redmiles, E. (2022). " nudes? shouldn't i charge for these?": Exploring what motivates content creation on onlyfans. *arXiv preprint arXiv:2205.10425*.

[34] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[35] Haylock, Z. (2020). Pornhub just deleted most of its content.

[36] Hernández-Álvarez, M. (2019). Detection of possible human trafficking in twitter. In *2019 International Conference on Information Systems and Software Technologies (ICI2ST)*, pages 187–191. IEEE.

[37] Hooi, B., Song, H. A., Beutel, A., Shah, N., Shin, K., and Faloutsos, C. (2016). Fraudar: Bounding graph fraud in the face of camouflage. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 895–904.

[38] Ian (2022). dbscan: Density-based spatial clustering of applications with noise (dbscan) and related algorithms.

[39] Jaadi, Z. (2022). A step-by-step explanation of principal component analysis (pca).

[40] Jiang, M., Cui, P., Beutel, A., Faloutsos, C., and Yang, S. (2016). Inferring lockstep behavior from connectivity pattern in large graphs. *Knowledge and Information Systems*, 48(2):399–428.

[41] Kadam, S. (2021). Generating word cloud in python.

[42] Khuller, S. and Saha, B. (2009). On finding dense subgraphs. In *International colloquium on automata, languages, and programming*, pages 597–608. Springer.

[43] Kulshrestha, A. (2021). *Detection of Organized Activity in misc Escort Advertisements*. McGill University (Canada).

[44] Kumar, S., Cheng, J., Leskovec, J., and Subrahmanian, V. (2017). An army of me: Sockpuppets in misc discussion communities. In *Proceedings of the 26th International Conference on World Wide Web*, pages 857–866.

[45] Lee, C., Grasso, C., and Sharlow, M. F. (2002). Multiple sequence alignment using partial order graphs. *Bioinformatics*, 18(3):452–464.

[46] Lee, M.-C., Vajiac, C., Kulshrestha, A., Levy, S., Park, N., Jones, C., Rabbany, R., and Faloutsos, C. (2021). Infoshield: generalizable information-theoretic human-trafficking detection. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, pages 1116–1127. IEEE.

[47] Leskovec, J. (2017). Graphsage: Inductive representation learning on large graphs.

[48] Li, L., Simek, O., Lai, A., Daggett, M., Dagli, C. K., and Jones, C. (2018). Detection and characterization of human trafficking networks using unsupervised scalable text template matching. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3111–3120. IEEE.

[49] Lloyd Trefethen, D. B. (2020). Singular value decomposition as simply as possible.

[50] López, C. (2020). A wave of people turned to onlyfans to earn money when they lost their jobs due to the pandemic. *Insider. https://www. insider. com/people-are-creating-onlyfans-accounts-after-losing-jobs-during-pandemic-2020-6*.

[51] Mensikova, A. and Mattmann, C. A. (2018). Ensemble sentiment analysis to identify human trafficking in web data. In *Workshop on Graph Techniques for Adversarial Activity Analytics (GTA 2018), Marina Del Rey, CA, USA*, pages 5–9.

[52] MINOR, S. D. (2015). A report on the use of technology to recruit, groom and sell domestic minor sex trafficking victims.

[53] Neumann, S. (2018). Bipartite stochastic block models with tiny clusters. *Advances in Neural Information Processing Systems*, 31.

[54] Pacheco, D., Hui, P.-M., Torres-Lugo, C., Truong, B. T., Flammini, A., and Menczer, F. (2021). Uncovering coordinated networks on social media: Methods and case studies. *ICWSM*, 21:455–466.

[55] Polaris (2021). 2020 us national human trafficking hotline statistics. https://polarisproject.org/2020-us-national-human-trafficking-hotline-statistics/.

[56] Rabbany, R., Bayani, D., and Dubrawski, A. (2018). Active search of connections for case building and combating human trafficking. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2120–2129.

[57] Ramchandani, P., Bastani, H., and Wyatt, E. (2021). Unmasking human trafficking risk in commercial sex supply chains with machine learning. *Available at SSRN 3866259*.

[58] Rita, L. (2020). Louvain algorithm.

[59] Rosvall, M. and Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the national academy of sciences*, 105(4):1118–1123.

[60] Sharma, K., Zhang, Y., Ferrara, E., and Liu, Y. (2020). Identifying coordinated accounts on social media through hidden influence and group behaviours. *arXiv preprint arXiv:2008.11308*.

[61] Shen, S., Radev, D., Patel, A., and Erkan, G. (2006). Adding syntax to dynamic programming for aligning comparable texts for the generation of paraphrases. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 747–754.

[62] Shin, K., Eliassi-Rad, T., and Faloutsos, C. (2016a). Corescope: Graph mining using k-core analysis—patterns, anomalies and algorithms. In *2016 IEEE 16th international conference on data mining (ICDM)*, pages 469–478. IEEE.

[63] Shin, K., Hooi, B., and Faloutsos, C. (2016b). M-zoom: Fast dense-block detection in tensors with quality guarantees. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 264–280. Springer.

[64] Shin, K., Hooi, B., Kim, J., and Faloutsos, C. (2017a). D-cube: Dense-block detection in terabyte-scale tensors. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 681–689.

[65] Shin, K., Hooi, B., Kim, J., and Faloutsos, C. (2017b). Densealert: Incremental dense-subtensor detection in tensor streams. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1057–1066.

[66] Simonson, E. (2021). Semi-supervised classification of social media posts: Identifying sex-industry posts to enable better support for those experiencing sex-trafficking. *arXiv preprint arXiv:2104.03233*.

[67] Synced (2021). Yann lecun hails msa transformer's 'huge progress' in protein contact prediction.

[68] Thomas (2020). The economics of onlyfans.

[69] Tong, E., Zadeh, A., Jones, C., and Morency, L.-P. (2017). Combating human trafficking with deep multimodal models. *arXiv preprint arXiv:1705.02735*.

[70] U.S. Department of State (2021). 2021 trafficking in persons report. https://www.state.gov/reports/2021-trafficking-in-persons-report/.

[71] Uttarapong, J., Bonifacio, R., Jereza, R., and Wohn, D. Y. (2022). Social support in digital patronage: Onlyfans adult content creators as an misc community. In *CHI Conference on Human Factors in Computing Systems Extended Abstracts*, pages 1–7.

[72] Vatsal (2022). Louvain's algorithm for community detection in python.

[73] Vlerick, M. and Van Hove, J. (2021). Revolutionising digital sex work: an analysis of the impact of onlyfans on sex workers.

[74] Wang, J., Levy, S., Wang, R., Kulshrestha, A., and Rabbany, R. (2019). Scg: Spotting coordinated groups in social media. *arXiv preprint arXiv:1910.07130*.

[75] Wei, L. H. (2022). 10 onlyfans statistics you need to know in 2022.

[76] Wheeler, F. . C. (2021). Federal human trafficking report.

[77] Xanadu (2020). Dense subgraphs.

[78] Zhang, D., Yin, J., Zhu, X., and Zhang, C. (2019). Attributed network embedding via subspace discovery. *Data Mining and Knowledge Discovery*, 33(6):1953–1980.

[79] Zhang, S., Zhou, D., Yildirim, M. Y., Alcorn, S., He, J., Davulcu, H., and Tong, H. (2017). Hidden: hierarchical dense subgraph detection with application to financial fraud detection. In *Proceedings of the 2017 SIAM International Conference on Data Mining*, pages 570–578. SIAM.

Appendix A – Appendix title

In this section we show the extended experiments. Before jumping into the extended experiments, we present the poster (see figure 31) containing an explanation of the Partial Intersection approach.

The below poster was presented during an online workshop hosted by Neurips. Here, since we only add my collaboration along the one of the professors that supervised me, we do not discuss the Joint clustering approach nor the synthetic experiments.

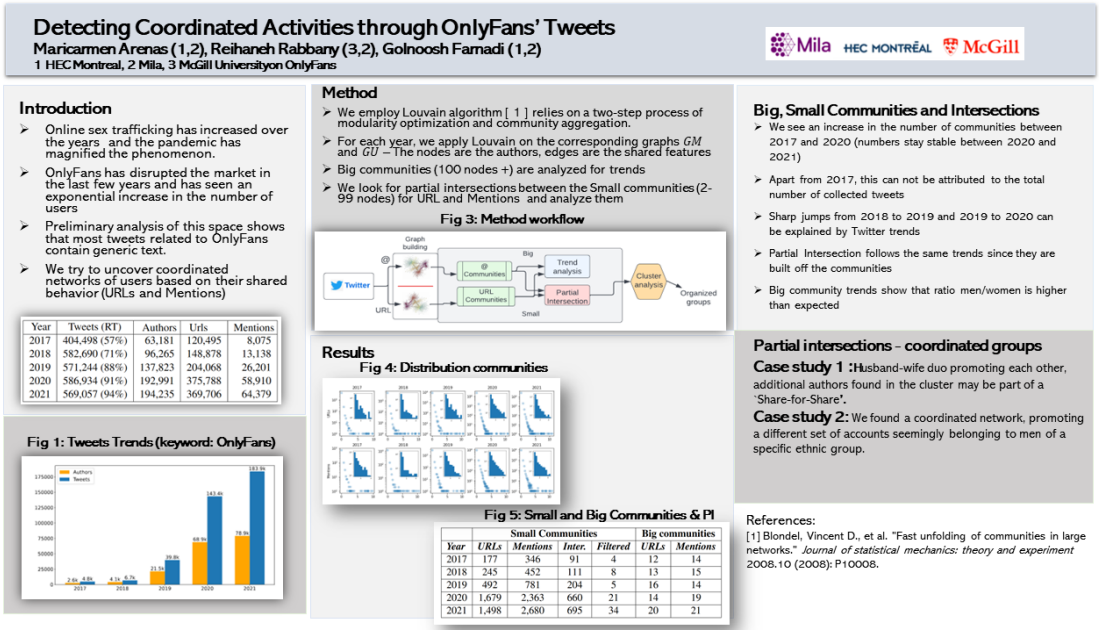


Figure 1: Poster presented in the WiML workshop at Neurips

Extended Experiments

In this chapter I showcase an extended set of results. These experiments helped establish the direction and steps of our article. It gave us a great amount of insight: it helped us understand our data and structure the pipeline of our method. Note that our Appendix was constructed before we started the experiments performed for our research paper, and was performed on a different dataset, therefore to provide a better understanding of these experiments, we kept all the steps we performed for this extended analysis.

Data collection

Our tweets are extracted with the help of the Twitter's API version 2, which was introduced by the end of 2020. This version also offers the 'Academic Research Track' which not only allows us to collect up to 10 million Tweets per month. Due to the monthly constraints, we set a cap of 10,000 tweets per day, so as to give us more time to reach our monthly quota. Note that this daily cap is reached somewhere in 2018. In other words, data for 2017 takes less time to collect than any other year. The following fields were collected in both file formats: author id, created_at, geo, id, lang, like_count, quote_count, reply_count,retweet_count, source, tweet. Note that the endpoint allows the user to extract a multitude of different fields, and we only chose restrictive amount. The most important fields to our research are: the Author id, the id (tweet id) and the text (tweeter text). Note that we can to choose the language, therefore we chose English. Hence, all the lines in the 'lang' column contain the word English. Furthermore, the only keyword provided to the query is **Onlyfans** but we exempt any tweet containing the words 'promotion' and 'promote' from our query since our previous analysis shows that these tweets are from individuals selling promotion services to other individuals having OnlyFans accounts.

Figures 32 and 33 present us with the number of tweets per year and per month respectively. We notice that the number of tweets more than doubles between years of 2017 to 2018, this increase can be explained by the fact that OnlyFans had around 100,000 users in 2017 and increased to 1 million users in 2018. Moreover, we also know that, while not

nearly as drastic, the number of Twitter users also increased during that period. On the other hand, we note that there was around 10% increase in tweets between years 2018 to 2019, but this is misleading since our data was capped to 10,000 per day and we more than likely reached the limit when approaching 2019. In fact, if we look at figure 33, we see that the number of tweets became steady around the 10th and 11th month of 2018. Let's also highlight that there were around 7 million OnlyFans users in 2019, which correspond to an 800% increase from 2018, and just as the year prior, twitter users increased steadily for the same period. Furthermore, we note that the number of tweets stay the same between years 2019 and 2021; which another important indicator that we reached the maximum daily tweeter collection. In short, we can not derive any conclusion regarding the growth of OnlyFans on Twitter for the periods of 2018 to 2021 based on this data collection. Note that, because of this, we were unable to study the effect of the pandemic on the platform, we therefore initiated a separate data collection done solely for the year 2020, this time, we assigned a cap of 1,000,000 tweets per day - note that due to time restrictions and data restrictions we were unable to collect tweets for the whole year.

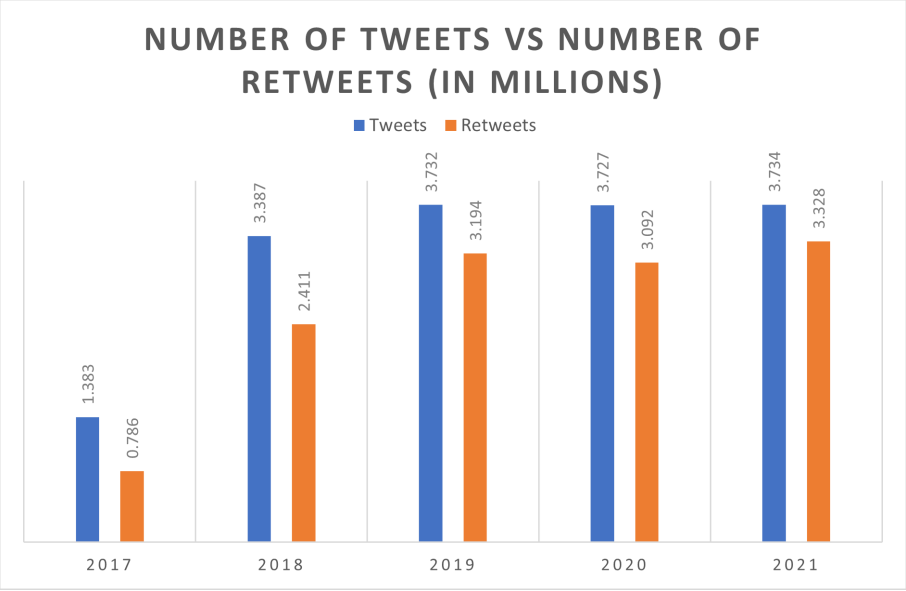


Figure 2: number of tweets vs retweets millions

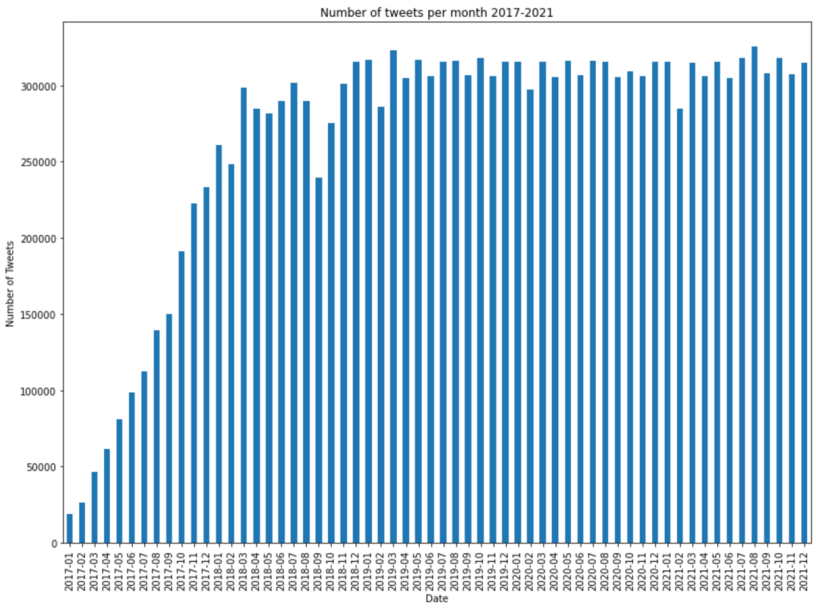


Figure 3: tweets per month

Regarding the type of tweets we collected, figure 32 shows us that most of our Tweets are actual Retweets. In fact, the number of Retweets increased dramatically between 2017 and 2018. In table 7 we see that Retweets represented 57% of the total tweets in 2017, while they represented 71% in 2018. The percentage of Retweet becomes stable from 2019 to 2021. This can be explained by the increased popularity of Onlyfans, if more people know about the platform and its creators, the posts are more likely to be shared.

Year	Percentage of Retweets
2017	57%
2018	71%
2019	86%
2020	83%
2021	89%

Table 1: Percentage of Retweets

Another important characteristic of our data is the number of authors. As we can see in figure 34 the number of users creating posts (authors) doubles from 2017 to 2018, and doubles once more from 2018 to 2019. On the other hand, we see year 2020 holds 1.6 times the number of authors found in 2019. The period between 2020 and 2021 is the only one where we see a decrease in the number of authors, however, this decrease is small. We note that the number of authors in 2017 to 2019 range from 8% to 13% of all tweets, while it reached 20% and 19% for years 2020 and 2021, respectively. We believe this is the reflection of the number of OnlyFans users and OnlyFans creators, which in both cases increases drastically during these period, and also the reflection of the number of tweets related to Onlyfans, which we weren't able to derive from our data collection.

Figure 4: number of authors per year

Our research focus on the text and authors, but it also focus on the URL and USERNAMES present in the tweets. Looking back at figure 32, where we show the number of Tweets per year and comparing it with figure 35, where we show the number of URLS and USERNAMES per year, we quickly conclude that there are generally more features than tweets. Here, we count only exception: the number of USERNAMES present in the 2017 tweets, in fact, here the number of USERNAMES represent 88% of the total number of tweets for that year. By contrast, there 130% more URLS present than there are tweets

on the same year. Moreover, this is also the only year where we see a large gap between the number of URLS and USERNAMES present. Figure 36, which represent the average number of URLS and USERNAMES per Tweet per year, shows us how for every following year the average features is above one, confirming that there are more features than tweets. This doesn't necessarily mean that every tweet contains one or more of one of both features, but it means that a percentage of our tweets contain multiple URLS or/and USERNAMES. It's also interesting to note that there are more URLS mentioned than USERNAMES both for 2017 and 2018, while there is an opposite scenario from 2019 to 2021. We do not have a clear idea of why there is such a difference, but it might be link to the manner in which the content creators are promoting their content.

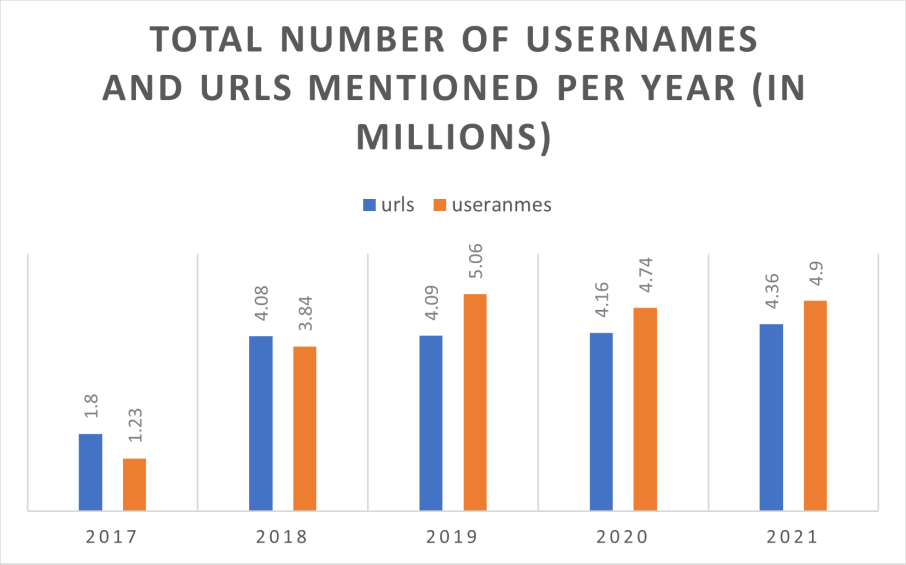


Figure 5: total number urls and users per year.

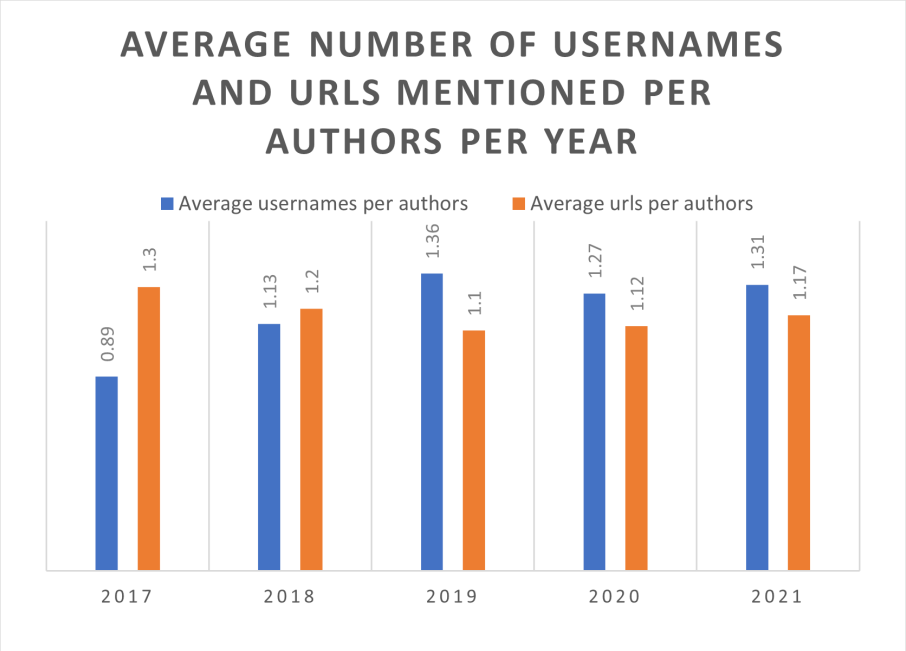


Figure 6: average number of users and urls per year

Twitter keywords

Since we need to understand the type of content our tweets have, we need to perform a Keyword analysis. We, of course, speculate that there will be the presence of sexual language, but it is important to proceed to a verification. To be clear, this is an important step because we are looking for people promoting OnlyFans accounts, we are not interested, for example, in posts stating opinions about OnlyFans. Note that this analysis took place while we were still collecting tweets from 2021, thus, here we present the results only for the period between 2017 to 2020.

Table 8 shows us the most common words, this is simply by counting the number of instances words appear in the whole dataset. Nota that, to get an accurate count, we applied lower case to each word. In addition to this step, we ignore stopwords from the count and only show the top 60 words. By looking at the results, we first notice that most of the words or emoticons (which are transformed into words) are implicit or explicit sexual language. On the other hand, many of the words such as watch, twitter, content, get, join, check, follow, fans, vid, video and subscribe, are keywords used to promote

Top words 1-20	Top words 21-40	Top words 41-60
watch	twitter	content
sweat_droplets	eggplant	sexy
big	get	today
join	co	check
ass	daddy	dick
guys	come	kiss_mark
videos	cock	amp
follow	hot	love
light_skin_tone	good	fuck
full	fans	fire
peach	one	subscribe
got	smiling_face_with_heart	like
day	smiling_face_with_horns	fun
make	vids	cum
time	winking_face_with_tongue	face_blowing_a_kiss
tongue	account	video
new	want	pics
page	see	rt

Table 2: List of top words

content. Note that the word 'onlyfans' is the 15th most popular while the expressions 'Https' and 'co' come 12th and 14th. This latter observation can be explained by the fact that these expressions are attached to URLs and since there are more URLs than Tweets, we are not surprise to find these characters among our list. On the other hand, the expression 'rt', that refers to 'retweet' is also on the top common words since around 60% of tweets are Retweets.

37 is a word class analysis we did based on the TF-IDF bigrams based on our yearly data. Here the bigger words carry more weight; they are more important. We also notice that the most important bigrams in 2017 and 2018 are related to Onlyfans promotion related vocabulary, we see that both years include expressions such as 'monthly subscription", 'exclusive content', 'unseen exclusive', 'fan get' and others that are of lesser value. On the other hand, years 2019 and 2020, while they still show bigrams related to promotion, we also see multiple bigrams containing words translated from emoticons. These

emoticons are for the most part conveying excitement and are often attached to the expression 'https', in other words, they are emphasising an URL. From this, we can conclude that 2019 and 2020 promote OnlyFans accounts in a different manner.

Figure 7: words cloud 2017-2020

The figure 38

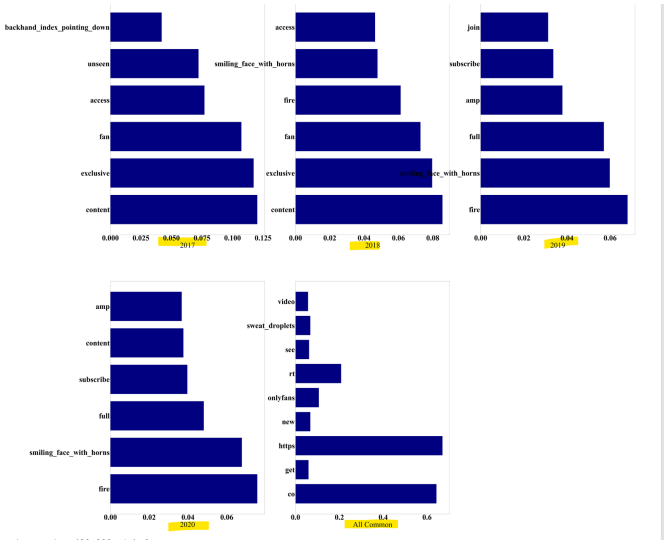


Figure 8: top words - based on TF-IDF- 2017-2020

Cluster evolution

Here we cluster our data based on the tweet's text using KMEANS. Note that the data is text-based, the tweets are transformed into a vector of numbers using Term Frequency-Inverse Document Frequency (TF-IDF). We use this simple method since we know that clustering based on text has already been applied in human trafficking research with positive results. Therefore we decided to apply the method on annual samples of tweets. In fact, we chose to apply KMEANS to 250,000 annual tweets, randomly selected from our dataset.

Here we do two different analysis, and we first start by analysing how clusters evolved through time based on a specific year. In our case we run two analysis, one for 2017 and one for 2021. For example, we take the year 2017 embedded sample data and fit the KMEANS model to this data, this will result in the clustering of our data for that year. Furthermore, it will compute the centers for that year. Because we want to see the evolution, we then apply the same centers (centroids) to the following years. We repeat the same steps but using year 2021 as our base. The results show that both models, based on 2017, as seen in figure 39, and based on 2021, as seen in40 are similar. They both have a predominant cluster that makes more than 90% of the total tweets and, for both years,

this dominant cluster becomes bigger through time. This confirms that our clusters do evolve through time. However, it fails to inform us on anything else. Therefore, further analysis need to be done.

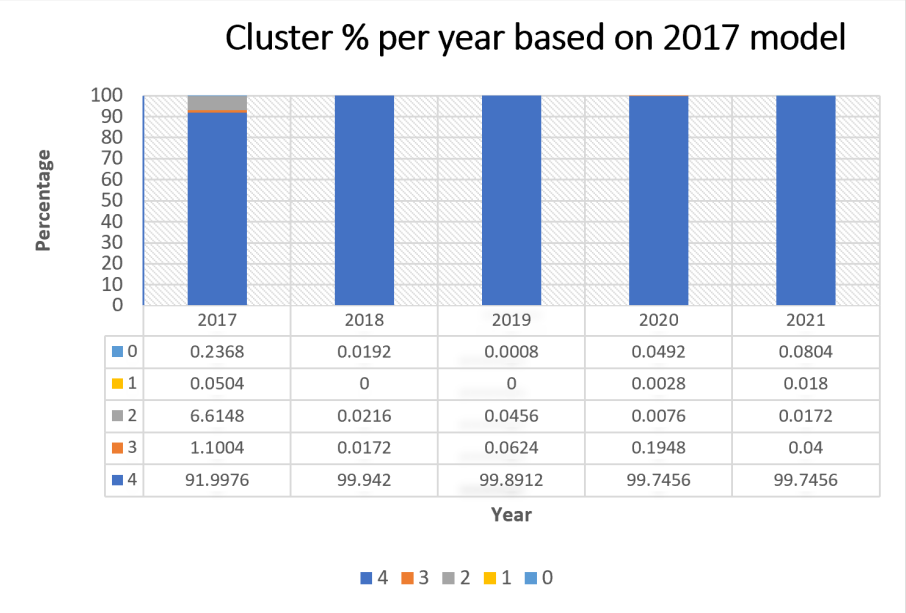
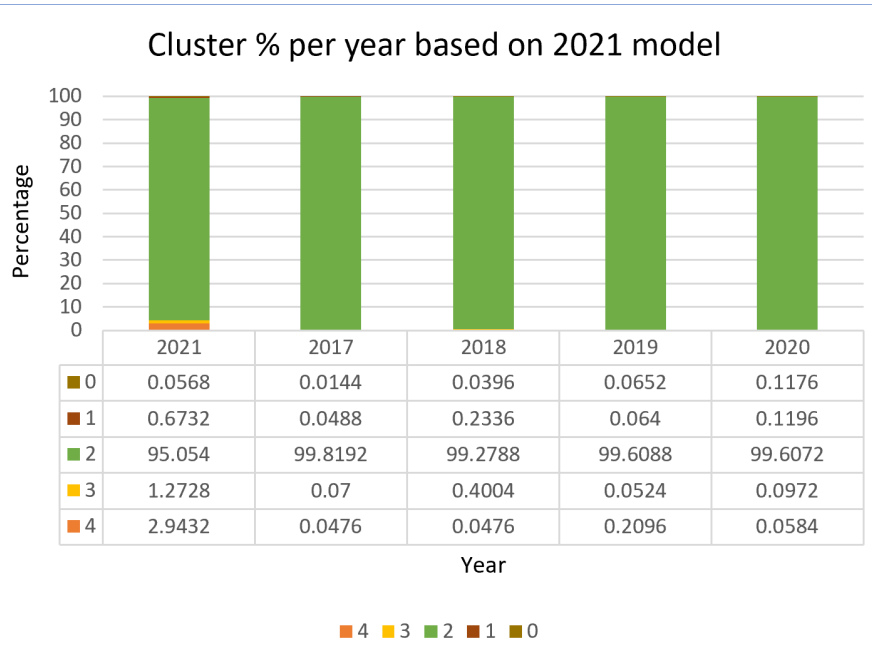


Figure 9: Clusters based on 2017



	2021	2017	2018	2019	2020
0	0.0568	0.0144	0.0396	0.0652	0.1176
1	0.6732	0.0488	0.2336	0.064	0.1196
2	95.054	99.8192	99.2788	99.6088	99.6072
3	1.2728	0.07	0.4004	0.0524	0.0972
4	2.9432	0.0476	0.0476	0.2096	0.0584

Figure 10: Clusters based on 2021

In addition to the prior analysis, we look specifically into the our clusters output from 2017. Note that this analysis is made separately, therefore the cluster's index number do not correspond to the images we have above. In fact, here the distribution is almost the same, there is a cluster containing around 90% of the tweets, in this case, it is cluster 0. Here we are left with two questions, do this clusters have a specific theme and why is cluster zero so big? To answer these questions we looked at the 5 tweets that are the closest to each of cluster's centroid. We can visualise the results in figure 41: in column one, we have the tweet's text, column two shows us the URL links extracted from the text, column three shows us the column to which each tweet is part of, and the last column shows us the distance between the tweet and the centroid of a cluster. By examining the clusters we note that, for example, cluster two contains error message regarding Twitter's policy, there are no URLS attached to it and they are all duplicates. If we look at the rest of the tweets, we cannot discern a specific theme since they are all promoting OnlyFans accounts, the only difference is the structure of the tweets, in other words, what differentiates them, is their 'template'.

2017 in detail – unigram Tweets per year

tweet	Tweet_urls	cluster	sqdist
@dimasgibi I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/6oiNoCQJyw ðŸ'ðŸ'	['https://t.co/6oiNoCQJyw']	0	0.08
Get access to my unseen and exclusive content at https://t.co/0OS7m8qiGA #findom	['https://t.co/0OS7m8qiGA']	0	0.09
"@sensualshades: RT @CandyKushXxX: Access to ALL my unseen and exclusive content at https://t.co/hM6YAmp9YP ðŸ˜˜ https://t.co/Hlp56AAUTt"	['https://t.co/hM6YAmp9YP', 'https://t.co/Hlp56	0	0.09
Slightly obsessed with the new project in case you had'nt guessed..There's just... https://t.co/Uac4afHSlA	['https://t.co/Uac4afHSlA']	0	0.1
RT @LaughingLatina: I've just got a new #fan! Get access to my unseen and exclusive content over 150 videos at https://t.co/HkQ1px97Bv httpâ€¦	['https://t.co/HkQ1px97Bv', 'httpâ€¦']	0	0.1
Just out the shower... https://t.co/SsO7DzY4tA	['https://t.co/SsO7DzY4tA']	1	0
#jordiporn My #fan u364212 has just sent me a $25.00 TIP! https://t.co/vhA8j4NO5x https://t.co/7I6DkneFIE	['https://t.co/vhA8j4NO5x', 'https://t.co/7I6Dkn	1	0
My #fan orlando02 has just sent me a $20.00 TIP! https://t.co/NfTYtco7c2 https://t.co/OQRC6eeCJu	['https://t.co/NfTYtco7c2', 'https://t.co/OQRC6e	1	0
RT @VictorialombaTV: New exclusive content, videos and photos in https://t.co/9hO5M51Ohz https://t.co/bfbbJsv9HW	['https://t.co/9hO5M51Ohz', 'https://t.co/bfbbJs	1	0
My #fan u1578458 has just sent me a $30.00 TIP! https://t.co/1mEWnDuMCE https://t.co/nmN86nu1a5	['https://t.co/1mEWnDuMCE', 'https://t.co/nmN	1	0
RT LaughingLatina "RT LaughingLatina: Another #fan just renewed their monthly subscription! See my exclusive conteâ€¦ https://t.co/ApWElMaVP7"	['https://t.co/ApWElMaVP7"']	2	0.17
RT LaughingLatina "RT LaughingLatina: Another #fan just renewed their monthly subscription! See my exclusive conteâ€¦ https://t.co/ApWElMaVP7"	['https://t.co/ApWElMaVP7"']	2	0.17
Another #fan just renewed their monthly subscription! ..Actually I had to lend him the money. He lives behind the Goodwill Store #onlyfans	[]	2	0.18
Another #fan has renewed their monthly subscription! See my exclusive and additional content at https://t.co/jFrCvcwgwH https://t.co/aoMNO918FL	['https://t.co/jFrCvcwgwH', 'https://t.co/aoMNO	2	0.19
Another #fan has renewed their monthly subscription! See my exclusive and additional content at https://t.co/jFrCvcwgwH https://t.co/8lxpVnREsM	['https://t.co/jFrCvcwgwH', 'https://t.co/8lxpVnF	2	0.19
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
This Tweet from @A_N_Ravenheart is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.69
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75

Figure 11: 20175clusterstweetsdetails

Since the overwhelming majority is inside cluster zero, we decided to look further into it. In fact, we proceed to add another KMEANS clustering on this specific cluster. After inspecting the cluster as seen in figure 42, we see that all the clusters contain the expression 'unseen and exclusive content' and they all have URL links attached to their corresponding tweets. Again, just like we noticed in the prior, every cluster seems to have its own template, in fact, they all contain duplicates or near duplicates per cluster.

5 Clusters derived from Cluster 0

tweet	Tweet_urls	cluster	sqdist
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/GJ4cgk2mTH https://t.co/xnEFxWxtJk	['https://t.co/GJ4cgk2mTH', 'https://t.co/xnEFxWxtJk']	0	0
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/4JXS8YU5ig https://t.co/NgHwdJmp6o	['https://t.co/4JXS8YU5ig', 'https://t.co/NgHwdJmp6o']	0	0
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/X4weWvmMre https://t.co/jpOKu0xWax	['https://t.co/X4weWvmMre', 'https://t.co/jpOKu0xWax']	0	0
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/OblNPz61WZ https://t.co/VqZP76SWV6	['https://t.co/OblNPz61WZ', 'https://t.co/VqZP76SWV6']	0	0
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/nrtROBQT3X https://t.co/axSnV8c1Pg	['https://t.co/nrtROBQT3X', 'https://t.co/axSnV8c1Pg']	0	0
Get access to my unseen and exclusive content at https://t.co/Q6ot3HXKbs	['https://t.co/Q6ot3HXKbs']	1	0.1
Get access to my unseen and exclusive content at; https://t.co/5MgtEoVOK4	['https://t.co/5MgtEoVOK4']	1	0.1
Get access to my unseen and exclusive content at https://t.co/0OS7m8qiGA #findom	['https://t.co/0OS7m8qiGA']	1	0.1
Get access to my unseen and exclusive content at https://t.co/S4fhbpJfrf by #Vdsxx1 via @ridder_co https://t.co/MYOwTlKtx1	['https://t.co/S4fhbpJfrf', 'https://t.co/MYOwTlKtx1']	1	0.12
Get access to @KatieGee032 unseen and exclusive content at https://t.co/RPO4RNDJoO ðŸ”¥â¤ï¸ðŸ”¥ https://t.co/Cgvf3huz2z	['https://t.co/RPO4RNDJoO', 'https://t.co/Cgvf3huz2z']	1	0.12
RT @ALETTAOCEANXXXX: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/h3hmxXXXyC https://t.co/vBaCLnâ€¦	['https://t.co/h3hmxXXXyC', 'https://t.co/vBaCLnâ€¦']	2	0.53
RT @ALETTAOCEANXXXX: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/h3hmxXXXyC https://t.co/wpWVnjâ€¦	['https://t.co/h3hmxXXXyC', 'https://t.co/wpWVnjâ€¦']	2	0.53
RT @ALETTAOCEANXXXX: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/h3hmxXXXyC https://t.co/dlCo60â€¦	['https://t.co/h3hmxXXXyC', 'https://t.co/dlCo60â€¦']	2	0.53
RT @ALETTAOCEANXXXX: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/h3hmxXXXyC https://t.co/mTOlnWâ€¦	['https://t.co/h3hmxXXXyC', 'https://t.co/mTOlnWâ€¦']	2	0.53
RT @ALETTAOCEANXXXX: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/h3hmxXXXyC https://t.co/6juvfYâ€¦	['https://t.co/h3hmxXXXyC', 'https://t.co/6juvfYâ€¦']	2	0.53
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/xsQxGgdTHj https://t.co/Lbs4dFsbCr	['https://t.co/xsQxGgdTHj', 'https://t.co/Lbs4dFsbCr']	3	0.07
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/xsQxGgdTHj https://t.co/cvoTSqN8BK	['https://t.co/xsQxGgdTHj', 'https://t.co/cvoTSqN8BK']	3	0.07
I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/xsQxGgdTHj https://t.co/icOvxXOfWg	['https://t.co/xsQxGgdTHj', 'https://t.co/icOvxXOfWg']	3	0.07
RT @jaydajusa: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/xsQxGgvuyR https://t.co/EWTYlcL7QA	['https://t.co/xsQxGgvuyR', 'https://t.co/EWTYlcL7QA']	3	0.1
RT @jaydajusa: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/xsQxGgvuyR https://t.co/EWTYlcL7QA	['https://t.co/xsQxGgvuyR', 'https://t.co/EWTYlcL7QA']	3	0.1
RT @MissJessaRhodes: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/uof504fvHV https://t.co/SHTbBVâ€¦	['https://t.co/uof504fvHV', 'https://t.co/SHTbBVâ€¦']	4	0.03
RT @x_summers: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/kwBNz7jdaf https://t.co/171d9y7Mcz	['https://t.co/kwBNz7jdaf', 'https://t.co/171d9y7Mcz']	4	0.03
RT @KellyDivine: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/gRFYmbIk1M https://t.co/F2izAOfhal	['https://t.co/gRFYmbIk1M', 'https://t.co/F2izAOfhal']	4	0.03
RT @DannyDxxx: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/Hg97fAaIyH https://t.co/pN2cZKn1Cz	['https://t.co/Hg97fAaIyH', 'https://t.co/pN2cZKn1Cz']	4	0.03
RT @SamiParkerxxx: I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/IZ8CtjCaKT https://t.co/ax98klVEâ€¦	['https://t.co/IZ8CtjCaKT', 'https://t.co/ax98klVEâ€¦']	4	0.03

Figure 12: 5clustersfromcluster0

Pandemic Trends

Here we want to measure the effect the pandemic has on OnlyFans through Twitter. For that purpose, we try to collect data for year 2020. Due to time constraints, we collected tweets from January the 1st 2020 to August the 8th 2020, therefore we have seven full months. Before analysing our data, we need to explain the context. First off, all our tweets are in English, and as we know, most tweets are from the US, thus it makes

sense that we base our observations on Covid-19 announcements made by the World Health Organization (WHO) and the United Sates. In fact, it is in January 30, 2020 that the Unites States declares to have its first case of coronavirus, simultaneously, the WHO declares the virus to be a Public Health Emergency of International Concern [5]. Then we see that, in March 11, 2020, Covid-19 is declared to be an official Pandemic https://www.cnn.com/2021/08/09/health/covid-19-pandemic-timeline-fast-facts/index.html. We can speculate most worldwide lockdowns might have started around this period. In fact, the lockdown in New-York [6] and in most Canadian provinces, including Quebec [7], started mid-March. When it comes to the duration of pandemic, we know that the lockdown lasted for months and even a whole year, in fact, it is only after the widespread vaccination of the population that activities slowly returned to normal. This is an important detail, since our data does only 5 months of content tweeted during the pandemic.

However, it is likely that online activity wouldn't have diminish as mush as we think after lockdowns and/or strict measures. In fact, we can easily state that the pandemic changed the way we work and live. People now rely on technology and social media more than ever.

In figure 55 we see that, there is a steady increase in the number of Onlyfan's related tweets between the January to April, on the other hand, we note that between April and August, there is the number of tweets stay steady.

[5] https://www.cnn.com/2021/08/09/health/covid-19-pandemic-timeline-fast-facts/index.html
[6] https://www.investopedia.com/historical-timeline-of-covid-19-in-new-york-city-5071986
[7] https://montreal.ctvnews.ca/covid-19-in-quebec-a-timeline-of-key-dates-and-events-1.4892912

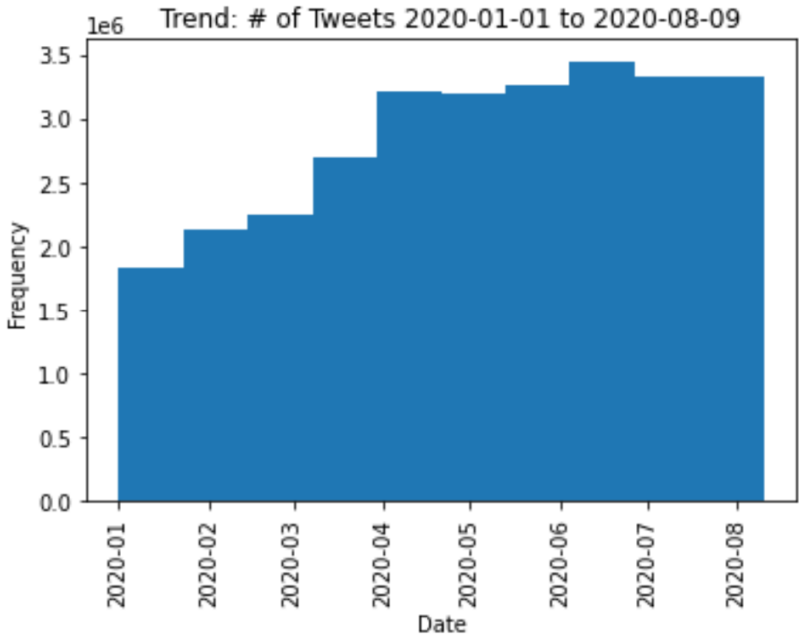


Figure 13: pandemic Trend 2020

Table **??**tweets between January and July 2020tab:Total OnlyFans tweets between January and July 2020ents us with a detailed numbers. In fact, the data shows us that there was a 10% increase between January and February, while there was a jump of 27% between February and March, the period between March and April sees an increase of 17%. Although May to July figures are more steady, we still see some lighter fluctuations, in fact, the there is a 8% increase between April and May, while there is a 5% increase between June and July. The only period where we see a decrease, it's between May and June, but this decrease is barely 3%. We can conclude from our data that Onlyfans was already growing (monthly) before the pandemic, and in fact, we see the biggest increase just right before the pandemic. However, the second biggest increase happens right after the pandemic. If we look at figure 44 [8], we see that OnlyFans Relative Search Volumes do have a sharp increase between February and May 2020. Then we see another steady increase starting from May 2020. Since this increase is constant, it makes sense that the search volumes were off the charts by January 2021. Furthermore, we know that

[8]https://www.thepourquoipas.com/post/the-case-for-an-onlyfans-ipo

Month - 2020	Number of Tweets
January	2,623,498
February	2,897,284
March	3,686,299
April	4,302,473
May	4,635,403
June	4,513,328
July	4,739,642

Table 3: Total Onlyfans tweets between January and July 2020

the number of users also increase substantially between this period. In fact, there are 12 million users in January 2020, while there are 30 million by May 2020, 75 million by October 2020, and 120 million by February 2021 [9]. In other words, the popularity of OnlyFans was built over a whole year, nevertheless, it is surprising to notice that tweets increased so slowly a month after the pandemic announcement. We believe that the situation can be explained by the fact that most of the tweets are promoted directly from the content creator or the people that manage their account, while there also has been a great increase in the number of OnlyFans content creators, these have not been as outstanding as the number of users. In fact, in May 2020, we only had 450 thousand content creators and this figure doubled only by the end of 2020. In other words, this increase was less sharp, and in fact, closer to the trend we found.

[9] https://thesmallbusinessblog.net/onlyfans-statistics/

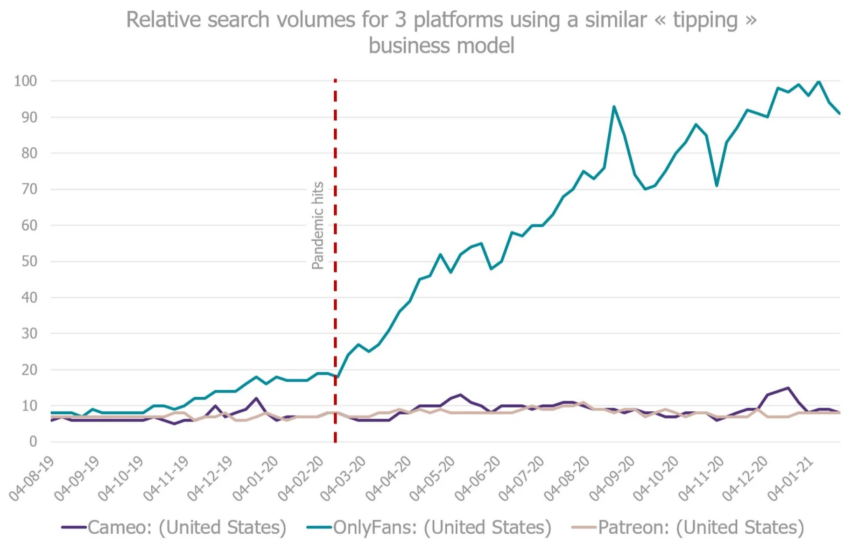


Figure 14: ONlyfansevolution

Lockstep Results

In this section we will discuss our lockstep method. Here we apply the Louvain Network detection method on URLS and on USERNAMES as well as the Infoshield Clustering method on the tweet's text. We then compare and investigate the results from the intersections (note that we use the exact same algorithm we have in our research paper see algorithm 1. Note that The intersect we investigate are from the smaller communities and clusters. In fact, according to paper (**?**), the authors note that human trafficking is more likely to be found in smaller clusters. To that we add that, big clusters and communities containing sex work advertisement trends that are not necessarily related to Human trafficking. This information, combined with the results above lead us to conduct our final coordinated behaviour analysis solely on the smaller communities and their associated Infoshield clusters.

Louvain matrices

As discussed in the methodology, we need to perform some transformations to our data before we apply the Louvain method. We first start with a yearly sample dataset, in our

case, we start with a sample of 250,000 tweets from 2017. From here we need tables author_id-USERNAMES and author_id-URL LINKS, these table must contain all the instance an url was associated to a specific author, this means if an author or a feature appears multiple times in the dataset, it will be also reflected in the table. Tables

	author_id	Urls	ReTweet	Pronoun
0	'2345826411	https://t.co/wSel6a0abk	1	0.0
1	'2345826411	https://t.co/fNy4y20GDv	1	0.0
2	'526867452	https://t.co/pac5B1z4U9	0	1.0
3	'161276101	https://t.co/d31H1SPOYc	0	0.0
4	'161276101	https://t.co/rzdvnVeOjo	0	0.0

Figure 15: table autor_id-URL

	author_id	Users	ReTweet	Pronoun
0	'2345826411	AllanrkXXX	1	0.0
1	'825556236615696385	AllanrkXXX	1	0.0
2	'169763107	AllanrkXXX	1	0.0
3	'897000714	AllanrkXXX	1	0.0
4	'789133018011017216	AllanrkXXX	1	0.0
...	...	...	...	...
1705006	'3317141718	Maximibd	1	1.0
1705007	'1145272478127808512	SashtheSoulsmit	0	0.0

Figure 16: table author_id-USERNAME

The autor_id-URL table contains 359,902 rows while the author_id-USERNAME table contains 354,898 rows. After this step is performed, we need to transpose our data (apply pivot), but since we need to keep only a specific amount of URLS and USERNAMES to count and order a unique URLS and USERNAMES table. This is important because we want to eliminate features appearing once or less in our, furthermore, we need to make sure the final matrix is light enough for server to handle.

To help us with our analysis we looked into the numbers of USERNAMES and URLS present in our dataset. Our dataset contains 95,883 (38%) tweets without a USERNAME,

Number of times	USERNAMES	URL LINKS
only 1	5,371	49,069
10 or more	2,470	4,725
25 or more	1,277	1,984
50 or more	715	984
100 or more	372	421

Table 4: USERNAMES and URLS instances

there are 13,058 unique ursernames while 29,247 (12%) of the tweets within our dataset do not contain URLS and there are 71,719 unique URLS. While there are more than double the amount of URLS than usernames, the table below shows that most of these URLS are only mentioned once (tweeted once) 10.

After we pivot our data we are left with dataframes resembling figure

author_id	http:https://t.co/H1Zb4	https//:https://t.co/L7	https://t.co/01EWqvSLMQ	https://t.co/01EyVQS0K5	https://t.co/01EyVR9C8F
'100007225	0.0	0.0	0.0	0.0	0.0
'1000126147	0.0	0.0	0.0	0.0	0.0
'1000297232	0.0	0.0	0.0	0.0	0.0
'1000408478	0.0	0.0	0.0	0.0	0.0
'1000462003	0.0	0.0	0.0	0.0	0.0

Figure 17: table transposed author_id-URLS

Note that analysis for the following years is different, but the idea behind it is the same, meaning that we need to make sure the matrices are of a certain size and that the features we keep are the ones appearing more than once.

URLS Louvain analysis results

As discussed in Chapter two, we ran louvain's dendrogram on the URL Graph, this resulted in 3 levels, meaning that the algorithm converges on the third level and that it produced three different partitions. The first level produced 2462 communities, while level two reached 943 communities, and the best partition produced 937 communities.

Here we note that level three only has six more partitions than level two, but we see a substantial difference between level one and the other levels.

In figure 48 below we can see the log distribution which contains the whole distribution, while figure 49 zooms into the distribution between nodes 1 and 75. Both figures represent best partition distribution, and in both we note that most of the communities only have one node. In fact, we have 684 communities consisting of only one node, which corresponds to 68% of all communities. Later in our analysis, we will discard those communities since these nodes are in actuality not part of any community.

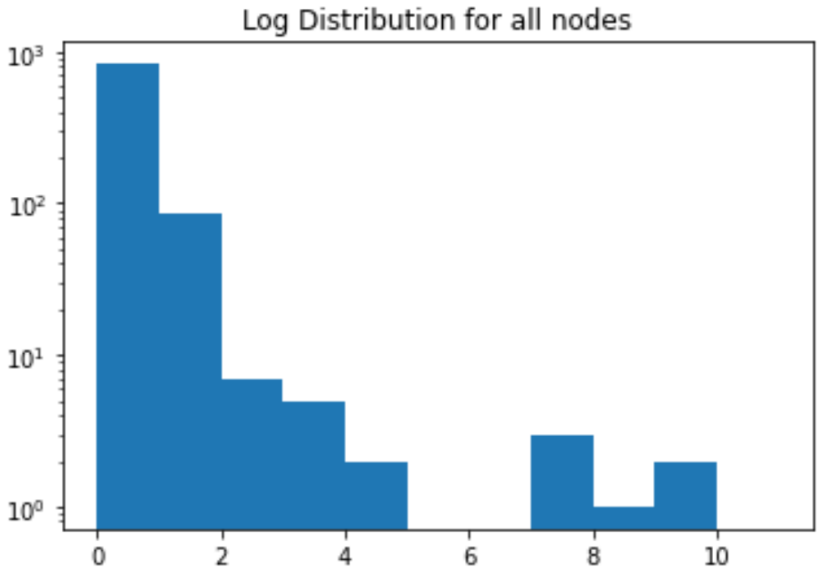


Figure 18: Log distribution of all nodes

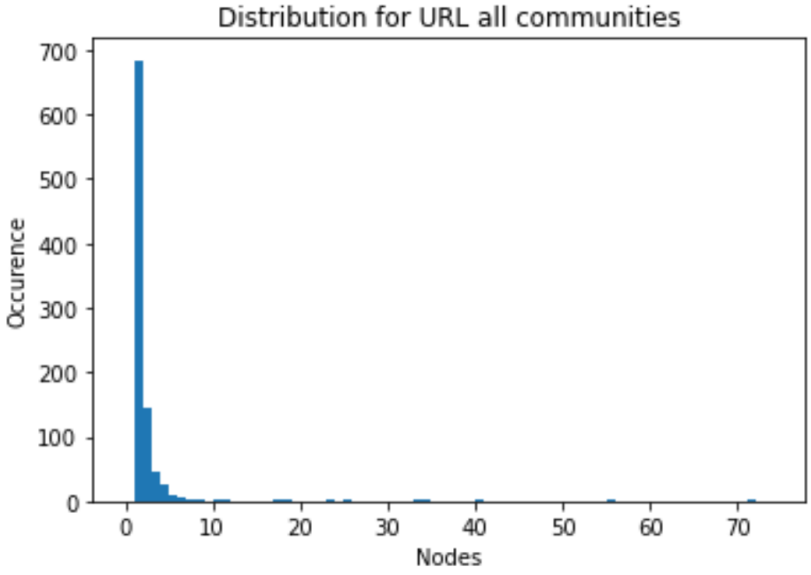


Figure 19: Distribution for nodes range 0 to 75

Furthermore, we note that our biggest community contains 17,078 nodes. Since there seems to be a huge difference between communities, we decided investigate which communities are the biggest. After exploring our data we decided to gather a group of communities having more than 100 nodes. We can see the results in table 12. In fact, here we are left with only 6 communities, meaning that there are 921 communities containing less than 100 nodes.

Community index	Community size
6	10491
233	2704
4	3375
7	17078
111	1234
11	2089

Table 5: Communities with 100 nodes and more URLs

At this point, we thought interesting to visualize the six communities, and since there are only a few groups, it will be easier to differentiate them. The Louvain-python package does have layouts to help us visualize our graph, however, in this case, it doesn't separate the communities properly, therefore we decided to use Gephi, a software specialised in graph visualization. As stated in chapter 2, Gephi can't work with too many nodes; 3000 is the approximate maximum. Therefore, to use Gephi, we have to remove the nodes that don't belong to communities with 100 nodes and more from the original graph, it is easier than creating a whole new graph. Our original graph contains 38,650 nodes, after removing the communities with 100 nodes and less, we are left with a graph containing 36,971. Since, this number is far higher than what we are allowed, we must take a percentage of each community, this will allow us to preserve the partition distribution. Technically, what we do, is remove close to 92% of nodes per community. Afterwards, we are left with a graph of containing 2841 nodes. In addition to this, we need to add the community index to the graph as an attribute to the node and save the graph. We can now open a project in Gephi and use the Atlas 2 template. After attributing a color to each community, we can visualize our graph. In image 55 we can see the six communities partitions and their occurrence.We note that nearly 75% of the output is made of two main communities.

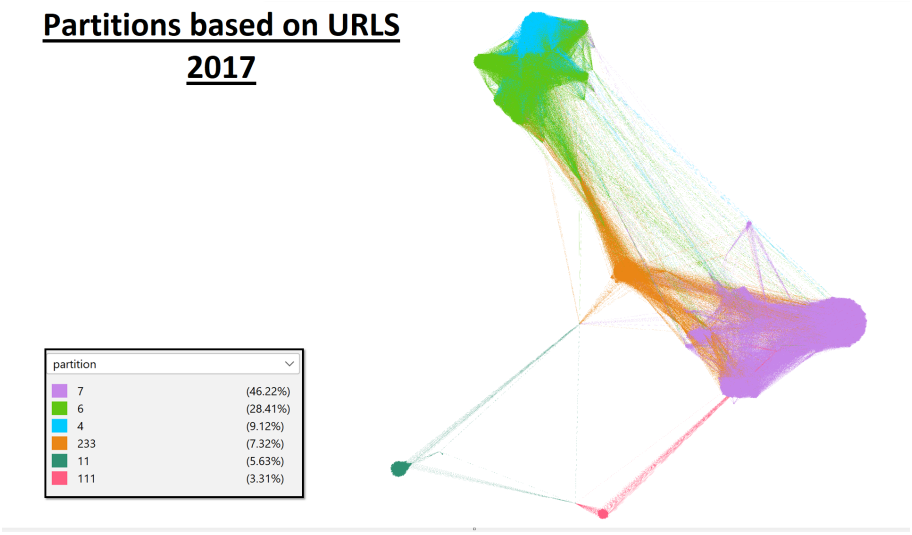


Figure 20: partitions graph based on urls 2017

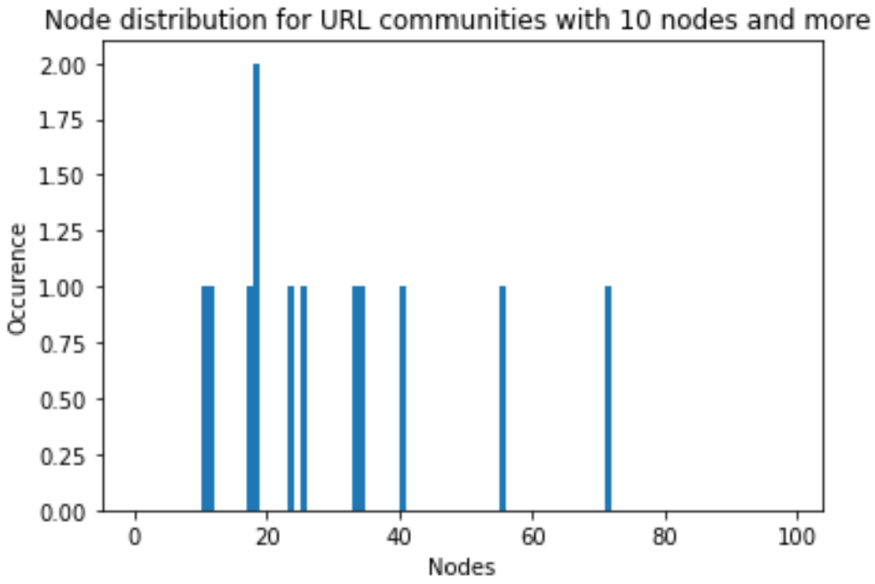


Figure 21: URL Distribution for 10 nodes and more

Our further analysis include smaller communities. We first look into communities containing 10 nodes or more. Since we already saw the distribution, we are not surprise to see that there are only 18 communities. Here in figure 52 we zoom into communities containing 10 nodes or more, we see that there is a disperse number of communities containing different number of nodes.

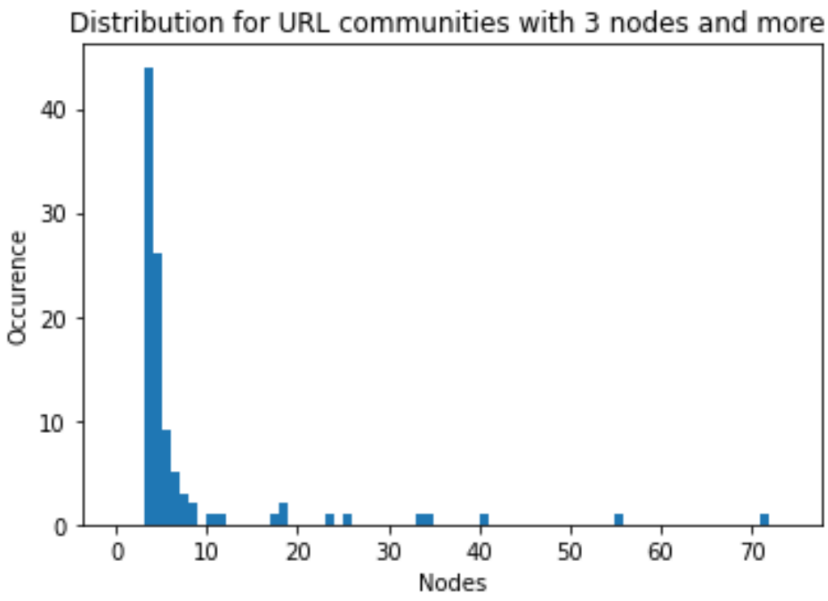


Figure 22: URL Distribution for 3 nodes and more

Secondly we look into communities containing 3 nodes or more. Here the number of communities reach 98, which is not that high of a number compared to the total number of communities in the partition. In image, 52 we zoom into communities with 3 nodes and more, and we note that most communities contain 3 to 5 nodes.

USERNAME Louvain analysis results

Here, we apply the same steps we followed in the previous section, therefore we start by producing our Graph G and run Louvain's algorithm to get a dendrogram as the main output. Here, the results shows only two levels: the first level has 2111 communities, while the best partition has 1397 communities. Here our biggest community contains 20,388 nodes. Since it will be near impossible to plot every single community, we create a graph that shows us the log distribution of all our nodes in our communities. Figure 53 is very similar to the one we saw for the URL analysis. Furthermore, if again, we zoom in into the communities having between 1 to 75, we note that 1158 of our communities consist of one node, which corresponds to 83% of our communities.

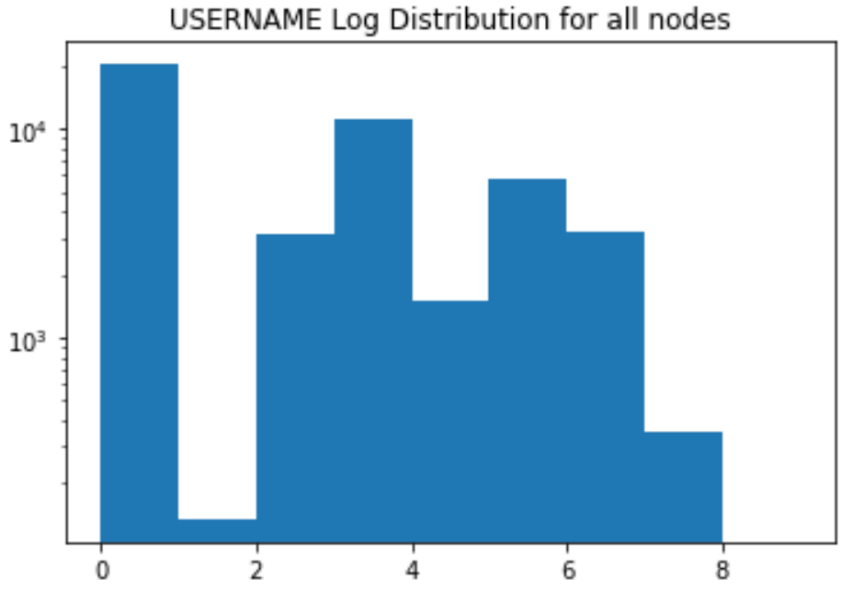


Figure 23: Log Distribution for USERNAME nodes

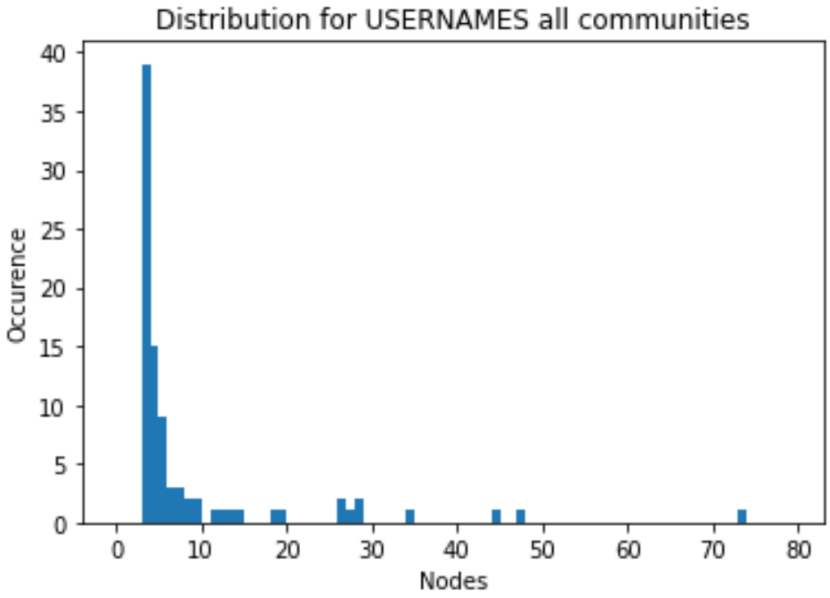


Figure 24: Distribution for USERNAME nodes range 1 to 75

Community index	Community size
24	10491
1	20388
47	430
373	4226
18	30044
89	1232
698	126
3	104
266	1110
492	2094

Table 6: Communities with 100 nodes and more

Just like we did for the Url analysis we created a new graph by removing the nodes that don't belong in partitions of 100 and more so we can use it to visualize our data. As mentioned before, since we are using Gephi can't work with more than 3000 nodes, so we have to reduce our graph. This is done by taking a percentage of each partition. We started with a graph containing 45,549 nodes, after removing the nodes allocated to paritions containing less than 100 nodes, we are left with 43,394 nodes. From here, we remove around 93% of the data per partition. We also remove all the nodes that are no longer part of those partitions from the graph. Now we are left with a graph containing 2,889 nodes.

Just as we did before, we need to add the partition number to the graph as an attribute to the node. Afterwards, we create a new project on Gephi and open the graph we just modified. Once again, we notice that we have two main communities, one is 47% of the graph and the other is nearly 25% of the graph.

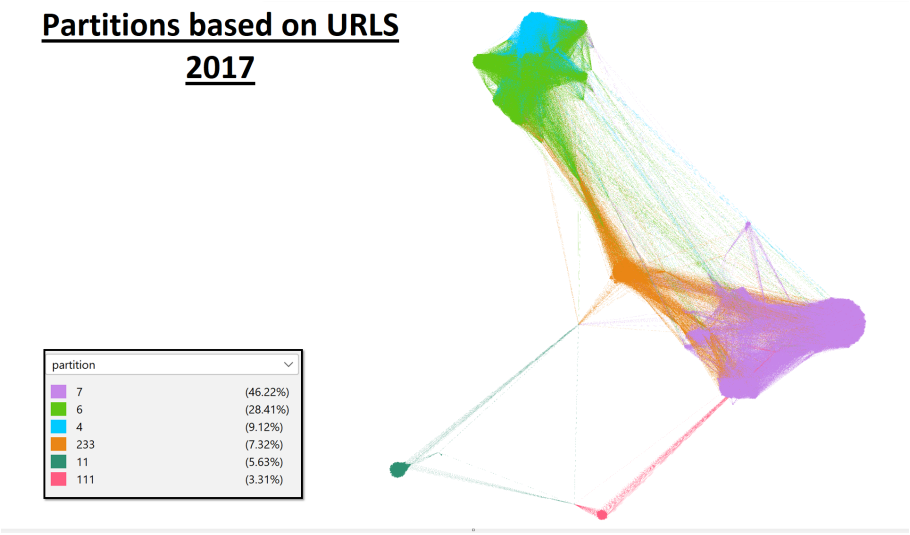


Figure 25: partitions graph based on urls 2017

Just like we did with our URL analysis, we examine our communities further by looking into communities that include less nodes. In fact, we look into communities having 10 nodes or more, the output shows us that we have 25 communities.In image 56 we zoom into this communities and we observe, just like in the URL analysis that our communities are more disperse. On the other hand, we have 98 communities with 3 nodes or more, and as we we can expect, most of the communities here do have 5 nodes or less, as we can see in figure 57.

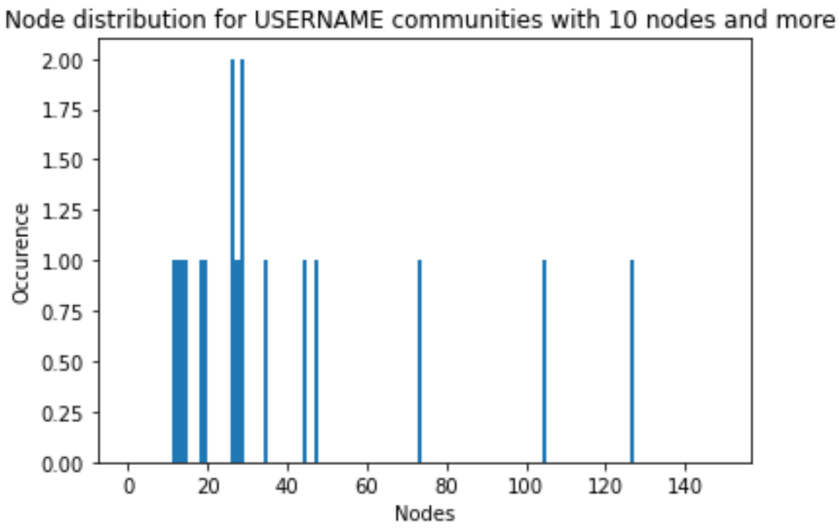


Figure 26: Distribution USERNAMES communities 10 and more

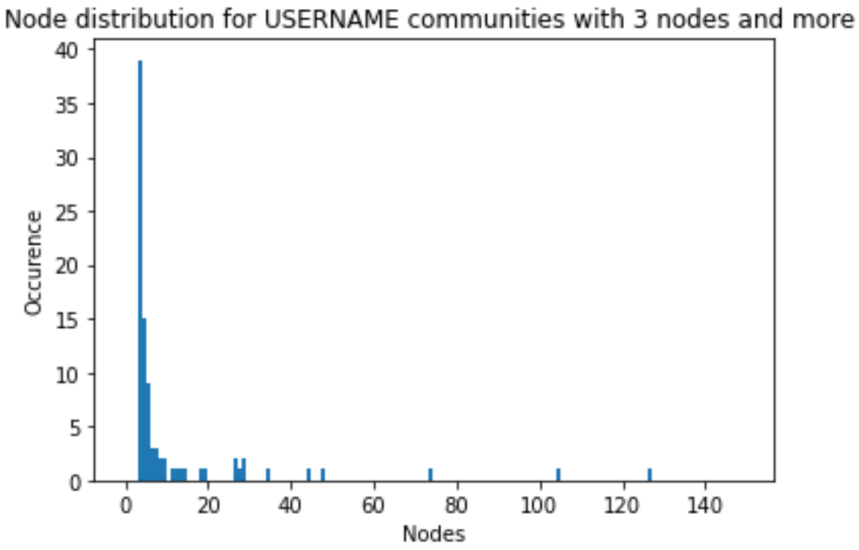


Figure 27: Distribution USERNAMES communities 3 and more

Comparison of Louvain URL and USERNAME

Here we want to compare our two previous analysis,namely the URL and the USERNAME Louvain community detection. We decided to apply two different methods of

comparison. First, we start by comparing both big communities, meaning both partitions containing communities of 100 nodes and more. This is done by iterating and comparing each group of node pertaining to a specific community in one feature analysis to the other. We are left with figure 58. The values in the matrix represent the total intersected nodes between both the URL and the USERNAME community. We added a simple map heat color code, only to show to show the extend of the overlapping when there is one. We added a percentage besides the values we deemed to be important intersects. This percentage is calculated by dividing the URL community size by the total number of nodes. Here we realize that for each URL community, we have an overlap of 85% and more with USERNAMES communities. We even have communities with 99% overlap. These results encouraged us to include the smaller communities to our analysis.

Confusion Matrix Best Partition 2017								
			URLS					
		Community Size	3375	10491	17078	2704	2089	1234
	Community Size	Community Number	1	2	3	4	5	6
USENAMES	20388	1	2	198 (4.4%)	16205 (95%)	233 (8.6%)	1	12
	2094	2	0	0	0	2	2081 (99.6%)	0
	104	3	0	58	33	0	0	0
	1110	4	1	703 (7%)	8	2	0	0
	430	5	0	4	238	0	0	0
	3004	6	0	4		2416 (89%)	0	0
	4226	7	3256 (96.5%)	463	5	1	0	0
	10680	8	108	8920 (85%)	35	10	0	0
	1232	9	0	0	14	0	0	1216 (99%)
	126	10	0	0	1	0	7	0

Figure 28: confusionMatrix2017

In short, we applied the normalized_mutual_info_score which was previously explained, to our both our partitions containing smaller communities. Note that we worked with communities of 3 nodes and more and 10 nodes and more. We discarded the rest simply because our further work won't include these smaller clusters. Since th normalized_mutual_info function compares two arrays that are ordered in the same manner, we mapped and united all our results in the same dataframe, in other words we mapped and ordered or data by merging dataframes together. For communities of 10 nodes our output was 0.77 while it was 0.72 for communities of 3 nodes and more. This was a satisfying result considering that it is in theory more difficult to overlap the smaller communities.

Our results gave us the confidence to pursue this lockstep venture by adding an additional observation: Infoshield. Furthermore, since our communities containing 3 nodes and more gave us a high score, we decided to continue or evaluation based on these communities. In this step, we also looked for the number of intersections between these two communities and find an overall of 44 intersections.

Comparison of Louvain with Infoshield

In this step we find and analyse the groups made of intersects between Louvain Communities and Infoshield. As discussed before, infoshield clusters the data based on its text. Consequently, each tweet has an Infoshield cluster number. On the other hand, each node or each author id, as a corresponding URL Louvain community and an USERNAME Louvain community. To compare them, we run Infoshield on dataframes that already have Louvain communities' assignments. Since there are two Louvain community analysis, we decided to run Louvain on both, we conclude that both Infoshield results will be very similar. Therefore, our choice will depend on further analysis of the data, however, both can be run. Note that we could have merged the communities results first and then proceed to apply the Infoshield method, but, in our case, a choice had to be made because of the way we chose to run the experiment (this is what we ended up doing in the paper).

Before starting the comparison of our three sets of groups, we need to verify if this step is necessary and feasible, therefore we apply the normalized_mutual_info_score on Communities and Infoshield. The comparison between the URL community versus the Infoshield community gives us a correlation of 0.18 while the USERNAME community versus the Infoshield community resulted in a correlation of 0.20. These correlations are not as strong as the ones gathered from solely the Communities comparison, however for the sake of testing, we also run these analysis on 'small' communities and 'big' communities separately. Here we consider that a big community contains 100 nodes and more while a small communities contains between 99 nodes and 3 nodes. The output of produced by the URL big communities and their associated Infoshield clusters is 0.17,

but the score produced by the smaller communities and the Infoshield clusters is slightly higher reaching 0.24. The results produced by the USERNAMES analysis are 0.19 and 0.47 respectively. As we note, the correlation between smaller communities and the Infoshield clusters is nearly 0.5, which is very high. Despite the lower correlation in respect to the URLs, we still believe it will be possible to find intersects between our different groups, however this is already giving us an indication as to say that Infoshield removes potentially interesting groups from our pool of intersections.

In fact, after running our simple algorithm presented in the previous chapter. Depending of which set of Infoshield cluster is chosen, we get 14 or 16 groups of intersects. Note that our code only takes in consideration groups that have at least 40% of their nodes in common (we are again using the Partial Intersection aglorithm found in the paper), that way we discard uninteresting groups from the start. Note that the analysis done with USERNAME Infoshield as two EXTRA groups, but all the other groups are the same as the ones found in the analysis using URL Infoshield. the The reason why it contains more groups of intersects is simply because the Infoshield clusters derived from USERNAMES showed more correlation with the USERNAME communities.

Figures 59 and 60 are the two outputs we have, each line gives the iteration number for the position of the group in the list it is iterating, for example, here we start with the URL community, so we get the iteration number then the community size, this is follow by the same information for the USERNAME community and Infoshield cluster groups. Underneath each line, we get the number of intersects. Note that sometimes this number is very small because we have a very small cluster. No matter the size of these groups of authors, they still might have multiple tweets which may contain multiple URLS or USERNAMES, therefore, they are still of interest for us.

```
intersecion between community # 1 4  and between community # 12 5 intersecion between cluster # 114 3
3
intersecion between community # 8 3  and between community # 96 4 intersecion between cluster # 122 3
3
intersecion between community # 24 18  and between community # 16 18 intersecion between cluster # 23 18
18
intersecion between community # 31 34  and between community # 21 34 intersecion between cluster # 97 33
33
intersecion between community # 33 4  and between community # 423 5 intersecion between cluster # 99 2
2
intersecion between community # 34 44  and between community # 31 33 intersecion between cluster # 84 32
31
intersecion between community # 40 4  and between community # 32 4 intersecion between cluster # 153 2
2
intersecion between community # 55 3  and between community # 52 3 intersecion between cluster # 146 2
2
intersecion between community # 64 7  and between community # 549 7 intersecion between cluster # 170 5
5
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 10 25
25
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 62 22
22
intersecion between community # 73 3  and between community # 292 4 intersecion between cluster # 112 2
2
intersecion between community # 87 3  and between community # 343 4 intersecion between cluster # 124 2
2
intersecion between community # 88 5  and between community # 346 5 intersecion between cluster # 132 5
5
total number of intersections: 14
```

Figure 29: Intersect 2017 based on URL Infoshield

```
intersecion between community # 1 4  and between community # 12 5 intersecion between cluster # 54 3
3
intersecion between community # 8 3  and between community # 96 4 intersecion between cluster # 57 3
3
intersecion between community # 24 18  and between community # 16 18 intersecion between cluster # 90 18
18
intersecion between community # 31 34  and between community # 21 34 intersecion between cluster # 61 33
33
intersecion between community # 33 4  and between community # 423 5 intersecion between cluster # 29 2
2
intersecion between community # 34 44  and between community # 31 33 intersecion between cluster # 13 31
31
intersecion between community # 40 4  and between community # 32 4 intersecion between cluster # 103 2
2
intersecion between community # 41 8  and between community # 812 8 intersecion between cluster # 49 13
8
intersecion between community # 55 3  and between community # 52 3 intersecion between cluster # 86 2
2
intersecion between community # 64 7  and between community # 549 7 intersecion between cluster # 94 5
5
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 39 22
22
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 96 25
25
intersecion between community # 73 3  and between community # 292 4 intersecion between cluster # 7 2
2
intersecion between community # 82 3  and between community # 691 4 intersecion between cluster # 23 3
2
intersecion between community # 87 3  and between community # 343 4 intersecion between cluster # 92 2
2
intersecion between community # 88 5  and between community # 346 5 intersecion between cluster # 10 5
5
total number of intersections: 16
```

Figure 30: Intersect 2017 based on USERNAME Infoshield

After getting our list of groups of intersects, we have to inspect each one and count how many unique URLS and unique USERNAMES they contain. For this purpose, we loop and filter our merged dataframe by each of the groups of intercept found. This process outputs two arrays. We filter our intersections by keeping the groups containing more than unique URL and USERNAMe.The only group filtered that we found contained 2 unique URLs and 3 unique USERNAMES. Note that seven out of the 16 intersections contained 3 or more USERNAMES but had only one unique URL. These are overall poor results. Based on this, we decided to do a new analysis without Infoshield, furthermore, our new analysis would be also done on communities containing 2 nodes (new analysis for the paper - we also ended up collecting new data).

Appendix A – Appendix title

In this section we show the extended experiments. Before jumping into the extended experiments, we present the poster (see figure 31) containing an explanation of the Partial Intersection approach.

The below poster was presented during an online workshop hosted by Neurips. Here, since we only add my collaboration along the one of the professors that supervised me, we do not discuss the Joint clustering approach nor the synthetic experiments.

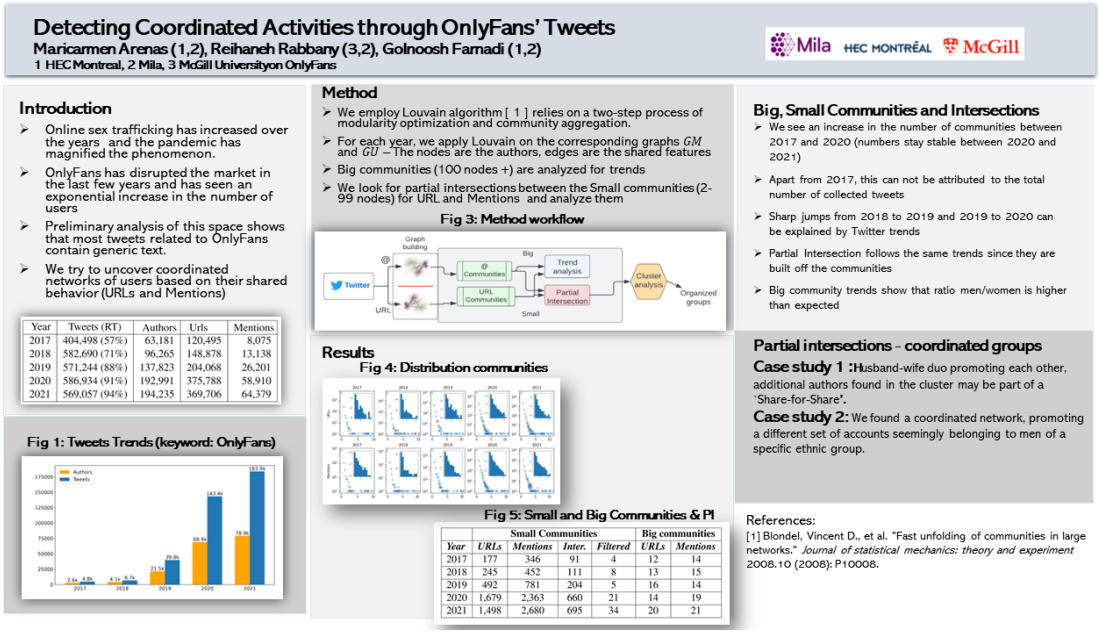


Figure 31: Poster presented in the WiML workshop at Neurips

Extended Experiments

In this chapter I showcase an extended set of results. These experiments helped establish the direction and steps of our article. It gave us a great amount of insight: it helped us understand our data and structure the pipeline of our method. Note that our Appendix was constructed before we started the experiments performed for our research paper, and was performed on a different dataset, therefore to provide a better understanding of these experiments, we kept all the steps we performed for this extended analysis.

Data collection

Our tweets are extracted with the help of the Twitter's API version 2, which was introduced by the end of 2020. This version also offers the 'Academic Research Track' which not only allows us to collect up to 10 million Tweets per month. Due to the monthly constraints, we set a cap of 10,000 tweets per day, so as to give us more time to reach our monthly quota. Note that this daily cap is reached somewhere in 2018. In other words, data for 2017 takes less time to collect than any other year. The following fields were collected in both file formats: author id, created_at, geo, id, lang, like_count, quote_count, reply_count,retweet_count, source, tweet. Note that the endpoint allows the user to extract a multitude of different fields, and we only chose restrictive amount. The most important fields to our research are: the Author id, the id (tweet id) and the text (tweeter text). Note that we can to choose the language, therefore we chose English. Hence, all the lines in the 'lang' column contain the word English. Furthermore, the only keyword provided to the query is **Onlyfans** but we exempt any tweet containing the words 'promotion' and 'promote' from our query since our previous analysis shows that these tweets are from individuals selling promotion services to other individuals having OnlyFans accounts.

Figures 32 and 33 present us with the number of tweets per year and per month respectively. We notice that the number of tweets more than doubles between years of 2017 to 2018, this increase can be explained by the fact that OnlyFans had around 100,000 users in 2017 and increased to 1 million users in 2018. Moreover, we also know that, while not

nearly as drastic, the number of Twitter users also increased during that period. On the other hand, we note that there was around 10% increase in tweets between years 2018 to 2019, but this is misleading since our data was capped to 10,000 per day and we more than likely reached the limit when approaching 2019. In fact, if we look at figure 33, we see that the number of tweets became steady around the 10th and 11th month of 2018. Let's also highlight that there were around 7 million OnlyFans users in 2019, which correspond to an 800% increase from 2018, and just as the year prior, twitter users increased steadily for the same period. Furthermore, we note that the number of tweets stay the same between years 2019 and 2021; which another important indicator that we reached the maximum daily tweeter collection. In short, we can not derive any conclusion regarding the growth of OnlyFans on Twitter for the periods of 2018 to 2021 based on this data collection. Note that, because of this, we were unable to study the effect of the pandemic on the platform, we therefore initiated a separate data collection done solely for the year 2020, this time, we assigned a cap of 1,000,000 tweets per day - note that due to time restrictions and data restrictions we were unable to collect tweets for the whole year.

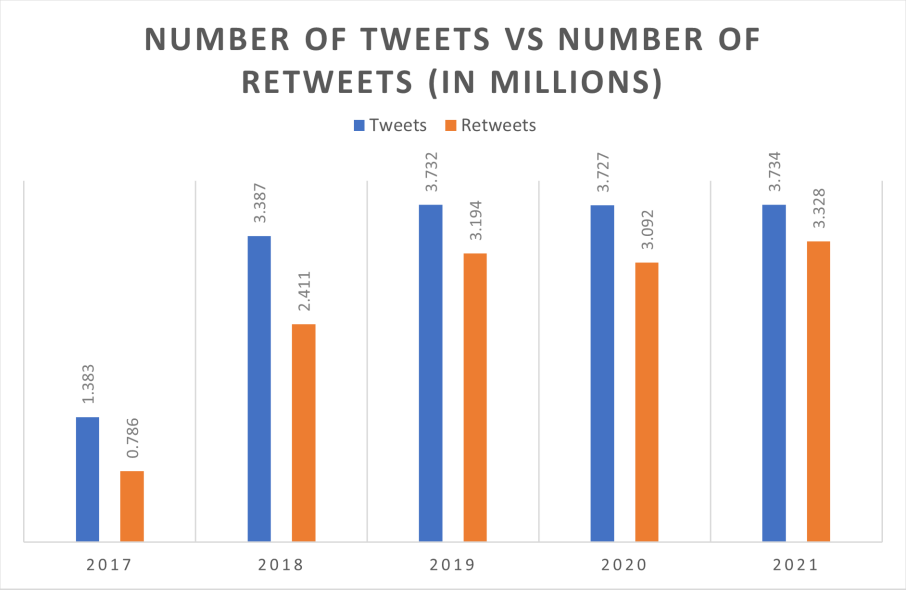


Figure 32: number of tweets vs retweets millions

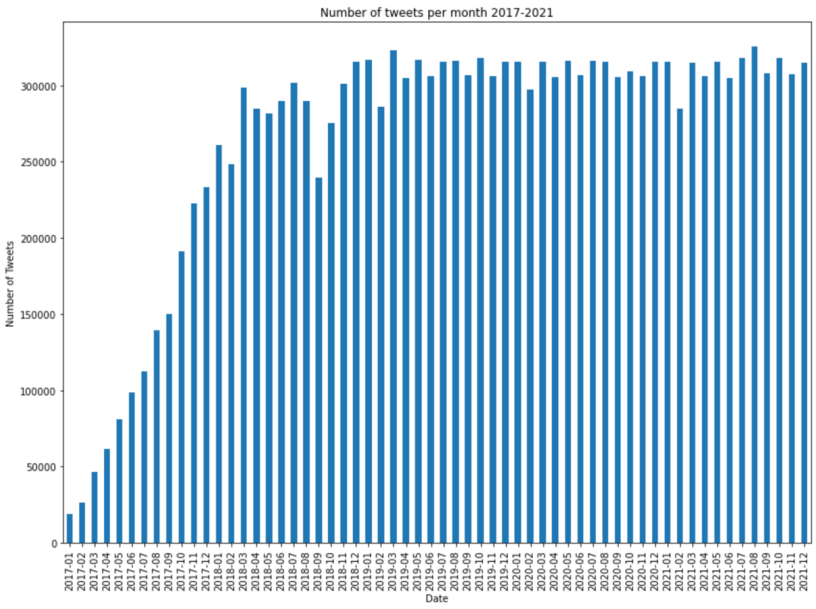


Figure 33: tweets per month

Regarding the type of tweets we collected, figure 32 shows us that most of our Tweets are actual Retweets. In fact, the number of Retweets increased dramatically between 2017 and 2018. In table 7 we see that Retweets represented 57% of the total tweets in 2017, while they represented 71% in 2018. The percentage of Retweet becomes stable from 2019 to 2021. This can be explained by the increased popularity of Onlyfans, if more people know about the platform and its creators, the posts are more likely to be shared.

Year	Percentage of Retweets
2017	57%
2018	71%
2019	86%
2020	83%
2021	89%

Table 7: Percentage of Retweets

Another important characteristic of our data is the number of authors. As we can see in figure 34 the number of users creating posts (authors) doubles from 2017 to 2018, and doubles once more from 2018 to 2019. On the other hand, we see year 2020 holds 1.6 times the number of authors found in 2019. The period between 2020 and 2021 is the only one where we see a decrease in the number of authors, however, this decrease is small. We note that the number of authors in 2017 to 2019 range from 8% to 13% of all tweets, while it reached 20% and 19% for years 2020 and 2021, respectively. We believe this is the reflection of the number of OnlyFans users and OnlyFans creators, which in both cases increases drastically during these period, and also the reflection of the number of tweets related to Onlyfans, which we weren't able to derive from our data collection.

Figure 34: number of authors per year

Our research focus on the text and authors, but it also focus on the URL and USERNAMES present in the tweets. Looking back at figure 32, where we show the number of Tweets per year and comparing it with figure 35, where we show the number of URLS and USERNAMES per year, we quickly conclude that there are generally more features than tweets. Here, we count only exception: the number of USERNAMES present in the 2017 tweets, in fact, here the number of USERNAMES represent 88% of the total number of tweets for that year. By contrast, there 130% more URLS present than there are tweets

on the same year. Moreover, this is also the only year where we see a large gap between the number of URLS and USERNAMES present. Figure 36, which represent the average number of URLS and USERNAMES per Tweet per year, shows us how for every following year the average features is above one, confirming that there are more features than tweets. This doesn't necessarily mean that every tweet contains one or more of one of both features, but it means that a percentage of our tweets contain multiple URLS or/and USERNAMES. It's also interesting to note that there are more URLS mentioned than USERNAMES both for 2017 and 2018, while there is an opposite scenario from 2019 to 2021. We do not have a clear idea of why there is such a difference, but it might be link to the manner in which the content creators are promoting their content.

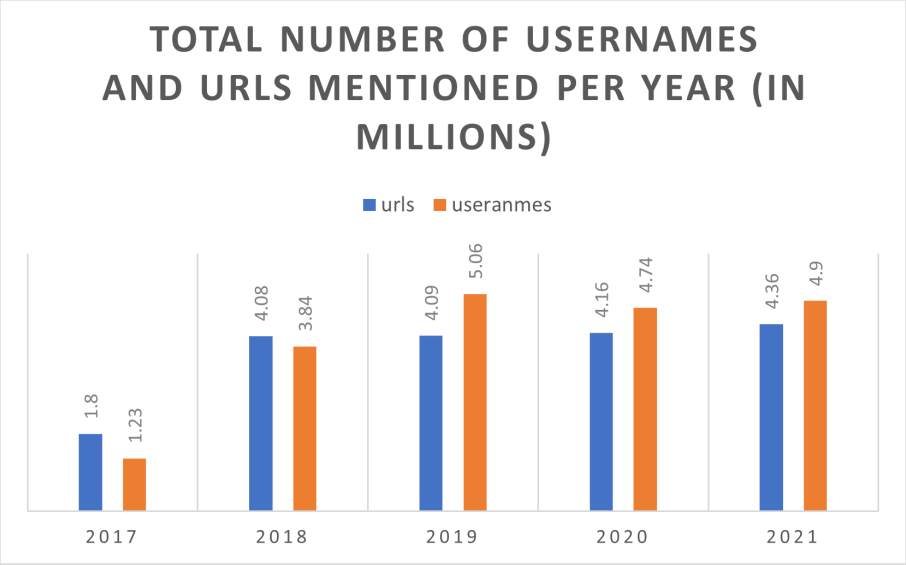


Figure 35: total number urls and users per year.

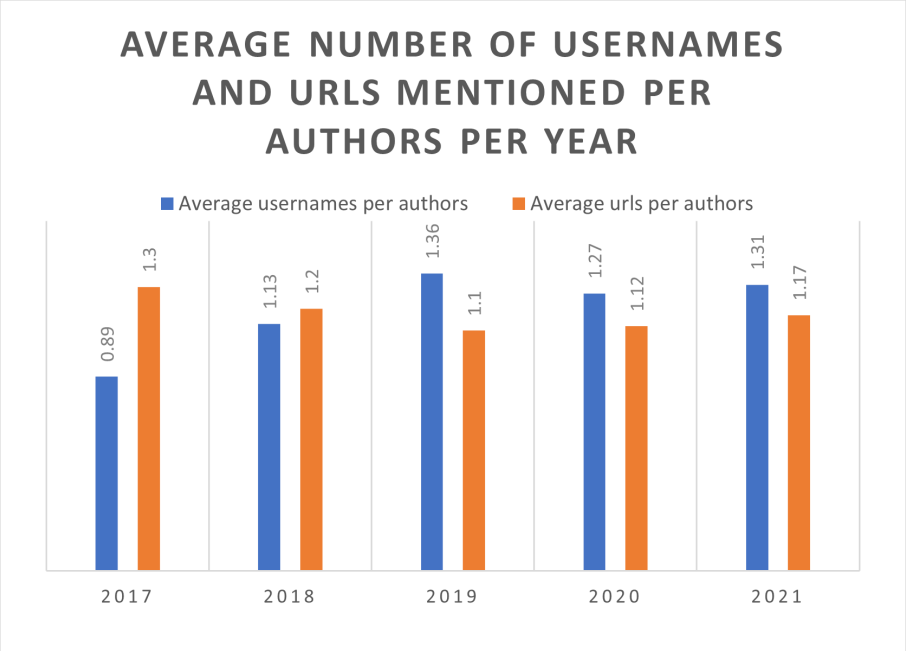


Figure 36: average number of users and urls per year

Twitter keywords

Since we need to understand the type of content our tweets have, we need to perform a Keyword analysis. We, of course, speculate that there will be the presence of sexual language, but it is important to proceed to a verification. To be clear, this is an important step because we are looking for people promoting OnlyFans accounts, we are not interested, for example, in posts stating opinions about OnlyFans. Note that this analysis took place while we were still collecting tweets from 2021, thus, here we present the results only for the period between 2017 to 2020.

Table 8 shows us the most common words, this is simply by counting the number of instances words appear in the whole dataset. Nota that, to get an accurate count, we applied lower case to each word. In addition to this step, we ignore stopwords from the count and only show the top 60 words. By looking at the results, we first notice that most of the words or emoticons (which are transformed into words) are implicit or explicit sexual language. On the other hand, many of the words such as watch, twitter, content, get, join, check, follow, fans, vid, video and subscribe, are keywords used to promote

Top words 1-20	Top words 21-40	Top words 41-60
watch	twitter	content
sweat_droplets	eggplant	sexy
big	get	today
join	co	check
ass	daddy	dick
guys	come	kiss_mark
videos	cock	amp
follow	hot	love
light_skin_tone	good	fuck
full	fans	fire
peach	one	subscribe
got	smiling_face_with_heart	like
day	smiling_face_with_horns	fun
make	vids	cum
time	winking_face_with_tongue	face_blowing_a_kiss
tongue	account	video
new	want	pics
page	see	rt

Table 8: List of top words

content. Note that the word 'onlyfans' is the 15th most popular while the expressions 'Https' and 'co' come 12th and 14th. This latter observation can be explained by the fact that these expressions are attached to URLs and since there are more URLs than Tweets, we are not surprise to find these characters among our list. On the other hand, the expression 'rt', that refers to 'retweet' is also on the top common words since around 60% of tweets are Retweets.

37 is a word class analysis we did based on the TF-IDF bigrams based on our yearly data. Here the bigger words carry more weight; they are more important. We also notice that the most important bigrams in 2017 and 2018 are related to Onlyfans promotion related vocabulary, we see that both years include expressions such as 'monthly subscription", 'exclusive content', 'unseen exclusive', 'fan get' and others that are of lesser value. On the other hand, years 2019 and 2020, while they still show bigrams related to promotion, we also see multiple bigrams containing words translated from emoticons. These

emoticons are for the most part conveying excitement and are often attached to the expression 'https', in other words, they are emphasising an URL. From this, we can conclude that 2019 and 2020 promote OnlyFans accounts in a different manner.

Figure 37: words cloud 2017-2020

The figure 38

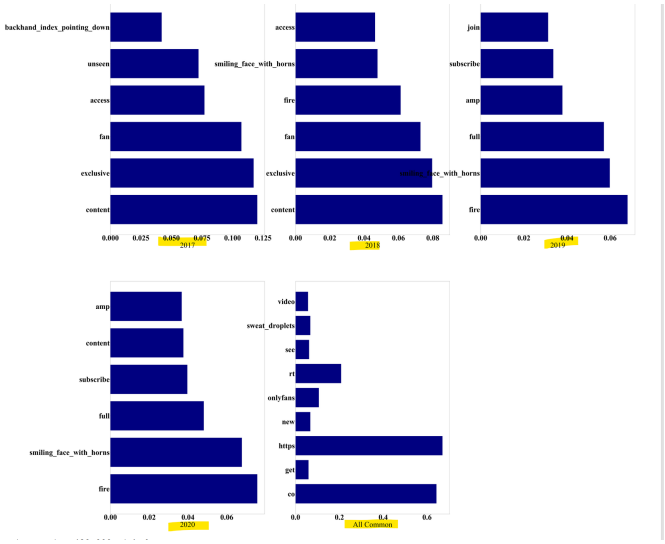


Figure 38: top words - based on TF-IDF- 2017-2020

Cluster evolution

Here we cluster our data based on the tweet's text using KMEANS. Note that the data is text-based, the tweets are transformed into a vector of numbers using Term Frequency-Inverse Document Frequency (TF-IDF). We use this simple method since we know that clustering based on text has already been applied in human trafficking research with positive results. Therefore we decided to apply the method on annual samples of tweets. In fact, we chose to apply KMEANS to 250,000 annual tweets, randomly selected from our dataset.

Here we do two different analysis, and we first start by analysing how clusters evolved through time based on a specific year. In our case we run two analysis, one for 2017 and one for 2021. For example, we take the year 2017 embedded sample data and fit the KMEANS model to this data, this will result in the clustering of our data for that year. Furthermore, it will compute the centers for that year. Because we want to see the evolution, we then apply the same centers (centroids) to the following years. We repeat the same steps but using year 2021 as our base. The results show that both models, based on 2017, as seen in figure 39, and based on 2021, as seen in40 are similar. They both have a predominant cluster that makes more than 90% of the total tweets and, for both years,

this dominant cluster becomes bigger through time. This confirms that our clusters do evolve through time. However, it fails to inform us on anything else. Therefore, further analysis need to be done.

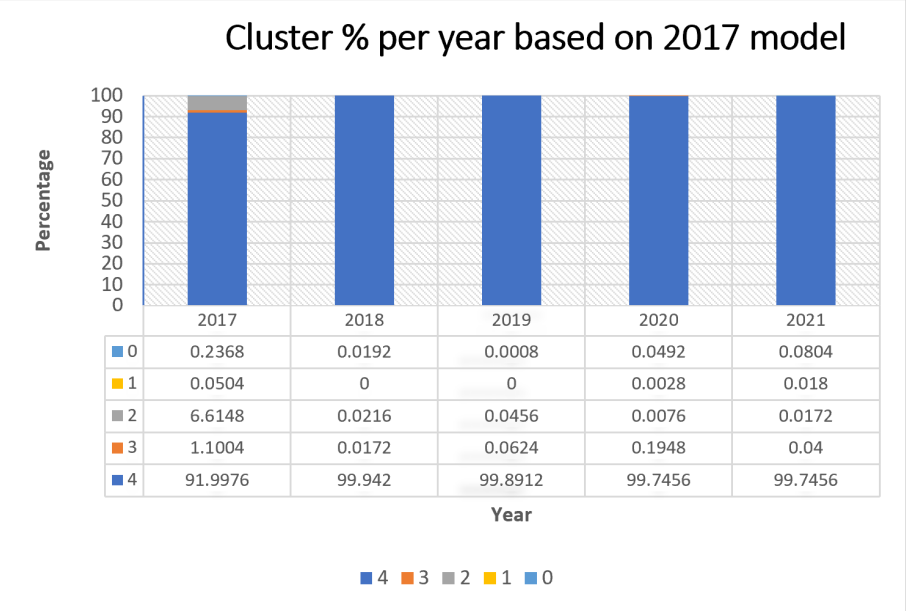
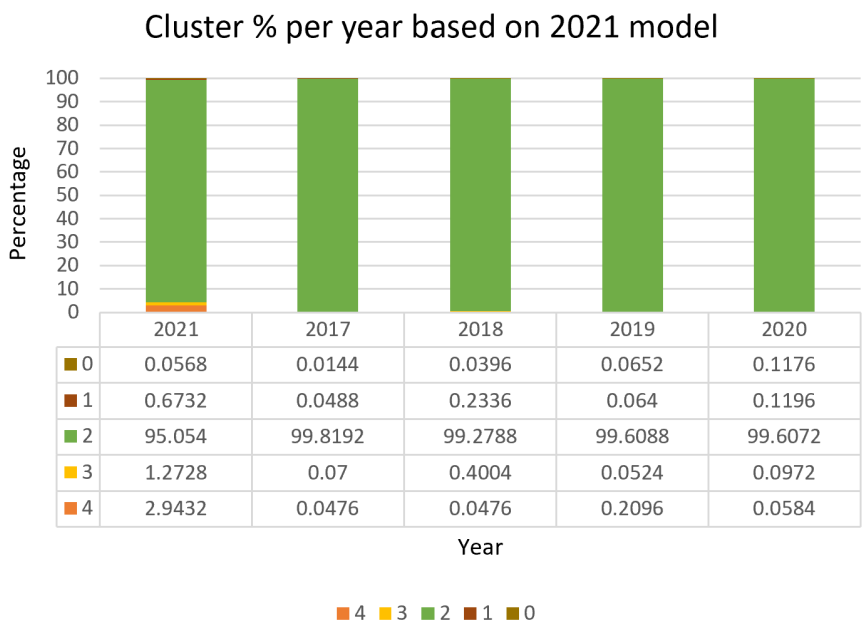


Figure 39: Clusters based on 2017



	2021	2017	2018	2019	2020
0	0.0568	0.0144	0.0396	0.0652	0.1176
1	0.6732	0.0488	0.2336	0.064	0.1196
2	95.054	99.8192	99.2788	99.6088	99.6072
3	1.2728	0.07	0.4004	0.0524	0.0972
4	2.9432	0.0476	0.0476	0.2096	0.0584

Figure 40: Clusters based on 2021

In addition to the prior analysis, we look specifically into the our clusters output from 2017. Note that this analysis is made separately, therefore the cluster's index number do not correspond to the images we have above. In fact, here the distribution is almost the same, there is a cluster containing around 90% of the tweets, in this case, it is cluster 0. Here we are left with two questions, do this clusters have a specific theme and why is cluster zero so big? To answer these questions we looked at the 5 tweets that are the closest to each of cluster's centroid. We can visualise the results in figure 41: in column one, we have the tweet's text, column two shows us the URL links extracted from the text, column three shows us the column to which each tweet is part of, and the last column shows us the distance between the tweet and the centroid of a cluster. By examining the clusters we note that, for example, cluster two contains error message regarding Twitter's policy, there are no URLS attached to it and they are all duplicates. If we look at the rest of the tweets, we cannot discern a specific theme since they are all promoting OnlyFans accounts, the only difference is the structure of the tweets, in other words, what differentiates them, is their 'template'.

2017 in detail – unigram Tweets per year

tweet	Tweet_urls	cluster	sqdist
@dimasgibi I've just got a new #fan! Get access to my unseen and exclusive content at https://t.co/6oiNoCQJyw ðŸ'ðŸ'	['https://t.co/6oiNoCQJyw']	0	0.08
Get access to my unseen and exclusive content at https://t.co/0OS7m8qiGA #findom	['https://t.co/0OS7m8qiGA']	0	0.09
"@sensualshades: RT @CandyKushXxX: Access to ALL my unseen and exclusive content at https://t.co/hM6YAmp9YP ðŸ˜˜ https://t.co/Hlp56AAUTt"	['https://t.co/hM6YAmp9YP', 'https://t.co/Hlp56	0	0.09
Slightly obsessed with the new project in case you had'nt guessed..There's just... https://t.co/Uac4afHSlA	['https://t.co/Uac4afHSlA']	0	0.1
RT @LaughingLatina: I've just got a new #fan! Get access to my unseen and exclusive content over 150 videos at https://t.co/HkQ1px97Bv httpâ€¦	['https://t.co/HkQ1px97Bv', 'httpâ€¦']	0	0.1
Just out the shower... https://t.co/SsO7DzY4tA	['https://t.co/SsO7DzY4tA']	1	0
#jordiporn My #fan u364212 has just sent me a $25.00 TIP! https://t.co/vhA8j4NO5x https://t.co/7I6DkneFIE	['https://t.co/vhA8j4NO5x', 'https://t.co/7I6Dkn	1	0
My #fan orlando02 has just sent me a $20.00 TIP! https://t.co/NfTYtco7c2 https://t.co/OQRC6eeCJu	['https://t.co/NfTYtco7c2', 'https://t.co/OQRC6e	1	0
RT @VictorialombaTV: New exclusive content, videos and photos in https://t.co/9hO5M51Ohz https://t.co/bfbbJsv9HW	['https://t.co/9hO5M51Ohz', 'https://t.co/bfbbJs	1	0
My #fan u1578458 has just sent me a $30.00 TIP! https://t.co/1mEWnDuMCE https://t.co/nmN86nu1a5	['https://t.co/1mEWnDuMCE', 'https://t.co/nmN	1	0
RT LaughingLatina "RT LaughingLatina: Another #fan just renewed their monthly subscription! See my exclusive conteâ€¦ https://t.co/ApWElMaVP7"	['https://t.co/ApWElMaVP7"']	2	0.17
RT LaughingLatina "RT LaughingLatina: Another #fan just renewed their monthly subscription! See my exclusive conteâ€¦ https://t.co/ApWElMaVP7"	['https://t.co/ApWElMaVP7"']	2	0.17
Another #fan just renewed their monthly subscription! ..Actually I had to lend him the money. He lives behind the Goodwill Store #onlyfans	[]	2	0.18
Another #fan has renewed their monthly subscription! See my exclusive and additional content at https://t.co/jFrCvcwgwH https://t.co/aoMNO918FL	['https://t.co/jFrCvcwgwH', 'https://t.co/aoMNO	2	0.19
Another #fan has renewed their monthly subscription! See my exclusive and additional content at https://t.co/jFrCvcwgwH https://t.co/8lxpVnREsM	['https://t.co/jFrCvcwgwH', 'https://t.co/8lxpVnF	2	0.19
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
RT @TSRileyQuinn: Join my @OnlyFansApp to see more exclusive content. https://t.co/jkMYwPDab1 https://t.co/OzwqBqn4D2	['https://t.co/jkMYwPDab1', 'https://t.co/OzwqE	3	0.07
This Tweet from @A_N_Ravenheart is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.69
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75
@ashelyrose95's account is temporarily unavailable because it violates the Twitter Media Policy. Learn more.	[]	4	0.75

Figure 41: 20175clusterstweetsdetails

Since the overwhelming majority is inside cluster zero, we decided to look further into it. In fact, we proceed to add another KMEANS clustering on this specific cluster. After inspecting the cluster as seen in figure 42, we see that all the clusters contain the expression 'unseen and exclusive content' and they all have URL links attached to their corresponding tweets. Again, just like we noticed in the prior, every cluster seems to have its own template, in fact, they all contain duplicates or near duplicates per cluster.

Figure 42: 5clustersfromcluster0

Pandemic Trends

Here we want to measure the effect the pandemic has on OnlyFans through Twitter. For that purpose, we try to collect data for year 2020. Due to time constraints, we collected tweets from January the 1st 2020 to August the 8th 2020, therefore we have seven full months. Before analysing our data, we need to explain the context. First off, all our tweets are in English, and as we know, most tweets are from the US, thus it makes

sense that we base our observations on Covid-19 announcements made by the World Health Organization (WHO) and the United Sates. In fact, it is in January 30, 2020 that the Unites States declares to have its first case of coronavirus, simultaneously, the WHO declares the virus to be a Public Health Emergency of International Concern [10]. Then we see that, in March 11, 2020, Covid-19 is declared to be an official Pandemic https://www.cnn.com/2021/08/09/health/covid-19-pandemic-timeline-fast-facts/index.html. We can speculate most worldwide lockdowns might have started around this period. In fact, the lockdown in New-York [11] and in most Canadian provinces, including Quebec [12], started mid-March. When it comes to the duration of pandemic, we know that the lockdown lasted for months and even a whole year, in fact, it is only after the widespread vaccination of the population that activities slowly returned to normal. This is an important detail, since our data does only 5 months of content tweeted during the pandemic.

However, it is likely that online activity wouldn't have diminish as mush as we think after lockdowns and/or strict measures. In fact, we can easily state that the pandemic changed the way we work and live. People now rely on technology and social media more than ever.

In figure 55 we see that, there is a steady increase in the number of Onlyfan's related tweets between the January to April, on the other hand, we note that between April and August, there is the number of tweets stay steady.

[10] https://www.cnn.com/2021/08/09/health/covid-19-pandemic-timeline-fast-facts/index.html
[11] https://www.investopedia.com/historical-timeline-of-covid-19-in-new-york-city-5071986
[12] https://montreal.ctvnews.ca/covid-19-in-quebec-a-timeline-of-key-dates-and-events-1.4892912

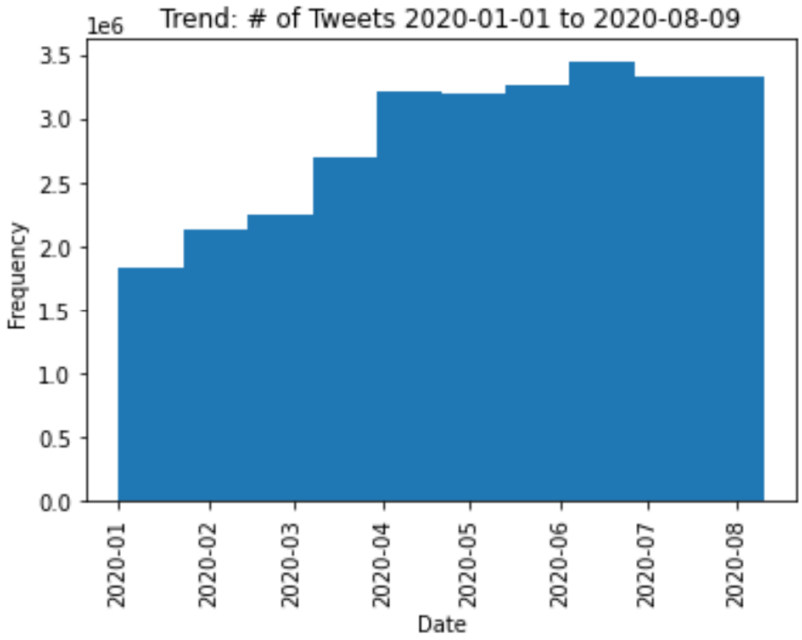


Figure 43: pandemic Trend 2020

Table **??**tweets between January and July 2020tab:Total OnlyFans tweets between January and July 2020ents us with a detailed numbers. In fact, the data shows us that there was a 10% increase between January and February, while there was a jump of 27% between February and March, the period between March and April sees an increase of 17%. Although May to July figures are more steady, we still see some lighter fluctuations, in fact, the there is a 8% increase between April and May, while there is a 5% increase between June and July. The only period where we see a decrease, it's between May and June, but this decrease is barely 3%. We can conclude from our data that Onlyfans was already growing (monthly) before the pandemic, and in fact, we see the biggest increase just right before the pandemic. However, the second biggest increase happens right after the pandemic. If we look at figure 44 [13], we see that OnlyFans Relative Search Volumes do have a sharp increase between February and May 2020. Then we see another steady increase starting from May 2020. Since this increase is constant, it makes sense that the search volumes were off the charts by January 2021. Furthermore, we know that

[13]https://www.thepourquoipas.com/post/the-case-for-an-onlyfans-ipo

Month - 2020	Number of Tweets
January	2,623,498
February	2,897,284
March	3,686,299
April	4,302,473
May	4,635,403
June	4,513,328
July	4,739,642

Table 9: Total Onlyfans tweets between January and July 2020

the number of users also increase substantially between this period. In fact, there are 12 million users in January 2020, while there are 30 million by May 2020, 75 million by October 2020, and 120 million by February 2021 [14]. In other words, the popularity of OnlyFans was built over a whole year, nevertheless, it is surprising to notice that tweets increased so slowly a month after the pandemic announcement. We believe that the situation can be explained by the fact that most of the tweets are promoted directly from the content creator or the people that manage their account, while there also has been a great increase in the number of OnlyFans content creators, these have not been as outstanding as the number of users. In fact, in May 2020, we only had 450 thousand content creators and this figure doubled only by the end of 2020. In other words, this increase was less sharp, and in fact, closer to the trend we found.

[14]https://thesmallbusinessblog.net/onlyfans-statistics/

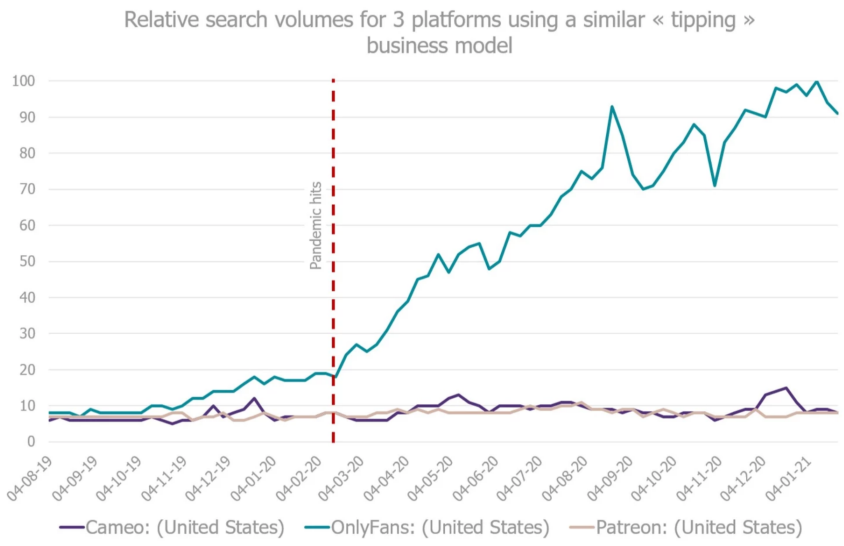


Figure 44: ONlyfansevolution

Lockstep Results

In this section we will discuss our lockstep method. Here we apply the Louvain Network detection method on URLS and on USERNAMES as well as the Infoshield Clustering method on the tweet's text. We then compare and investigate the results from the intersections (note that we use the exact same algorithm we have in our research paper see algorithm 1. Note that The intersect we investigate are from the smaller communities and clusters. In fact, according to paper (**?**), the authors note that human trafficking is more likely to be found in smaller clusters. To that we add that, big clusters and communities containing sex work advertisement trends that are not necessarily related to Human trafficking. This information, combined with the results above lead us to conduct our final coordinated behaviour analysis solely on the smaller communities and their associated Infoshield clusters.

Louvain matrices

As discussed in the methodology, we need to perform some transformations to our data before we apply the Louvain method. We first start with a yearly sample dataset, in our

case, we start with a sample of 250,000 tweets from 2017. From here we need tables author_id-USERNAMES and author_id-URL LINKS, these table must contain all the instance an url was associated to a specific author, this means if an author or a feature appears multiple times in the dataset, it will be also reflected in the table. Tables

	author_id	Urls	ReTweet	Pronoun
0	'2345826411	https://t.co/wSel6a0abk	1	0.0
1	'2345826411	https://t.co/fNy4y20GDv	1	0.0
2	'526867452	https://t.co/pac5B1z4U9	0	1.0
3	'161276101	https://t.co/d31H1SPOYc	0	0.0
4	'161276101	https://t.co/rzdvnVeOjo	0	0.0

Figure 45: table autor_id-URL

	author_id	Users	ReTweet	Pronoun
0	'2345826411	AllanrkXXX	1	0.0
1	'825556236615696385	AllanrkXXX	1	0.0
2	'169763107	AllanrkXXX	1	0.0
3	'897000714	AllanrkXXX	1	0.0
4	'789133018011017216	AllanrkXXX	1	0.0
...	...	...	...	...
1705006	'3317141718	Maximibd	1	1.0
1705007	'1145272478127808512	SashtheSoulsmit	0	0.0

Figure 46: table author_id-USERNAME

The autor_id-URL table contains 359,902 rows while the author_id-USERNAME table contains 354,898 rows. After this step is performed, we need to transpose our data (apply pivot), but since we need to keep only a specific amount of URLS and USERNAMES to count and order a unique URLS and USERNAMES table. This is important because we want to eliminate features appearing once or less in our, furthermore, we need to make sure the final matrix is light enough for server to handle.

To help us with our analysis we looked into the numbers of USERNAMES and URLS present in our dataset. Our dataset contains 95,883 (38%) tweets without a USERNAME,

Number of times	USERNAMES	URL LINKS
only 1	5,371	49,069
10 or more	2,470	4,725
25 or more	1,277	1,984
50 or more	715	984
100 or more	372	421

Table 10: USERNAMES and URLS instances

there are 13,058 unique ursernames while 29,247 (12%) of the tweets within our dataset do not contain URLS and there are 71,719 unique URLS. While there are more than double the amount of URLS than usernames, the table below shows that most of these URLS are only mentioned once (tweeted once) 10.

After we pivot our data we are left with dataframes resembling figure

author_id	http:https://t.co/H1Zb4	https//:https://t.co/L7	https://t.co/01EWqvSLMQ	https://t.co/01EyVQS0K5	https://t.co/01EyVR9C8F
'100007225	0.0	0.0	0.0	0.0	0.0
'1000126147	0.0	0.0	0.0	0.0	0.0
'1000297232	0.0	0.0	0.0	0.0	0.0
'1000408478	0.0	0.0	0.0	0.0	0.0
'1000462003	0.0	0.0	0.0	0.0	0.0

Figure 47: table transposed author_id-URLS

Note that analysis for the following years is different, but the idea behind it is the same, meaning that we need to make sure the matrices are of a certain size and that the features we keep are the ones appearing more than once.

URLS Louvain analysis results

As discussed in Chapter two, we ran louvain's dendrogram on the URL Graph, this resulted in 3 levels, meaning that the algorithm converges on the third level and that it produced three different partitions. The first level produced 2462 communities, while level two reached 943 communities, and the best partition produced 937 communities.

Here we note that level three only has six more partitions than level two, but we see a substantial difference between level one and the other levels.

In figure 48 below we can see the log distribution which contains the whole distribution, while figure 49 zooms into the distribution between nodes 1 and 75. Both figures represent best partition distribution, and in both we note that most of the communities only have one node. In fact, we have 684 communities consisting of only one node, which corresponds to 68% of all communities. Later in our analysis, we will discard those communities since these nodes are in actuality not part of any community.

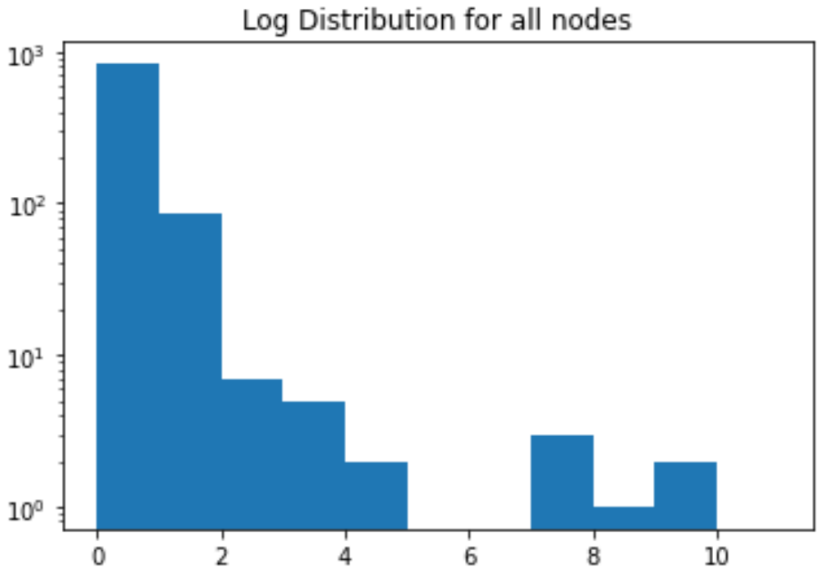


Figure 48: Log distribution of all nodes

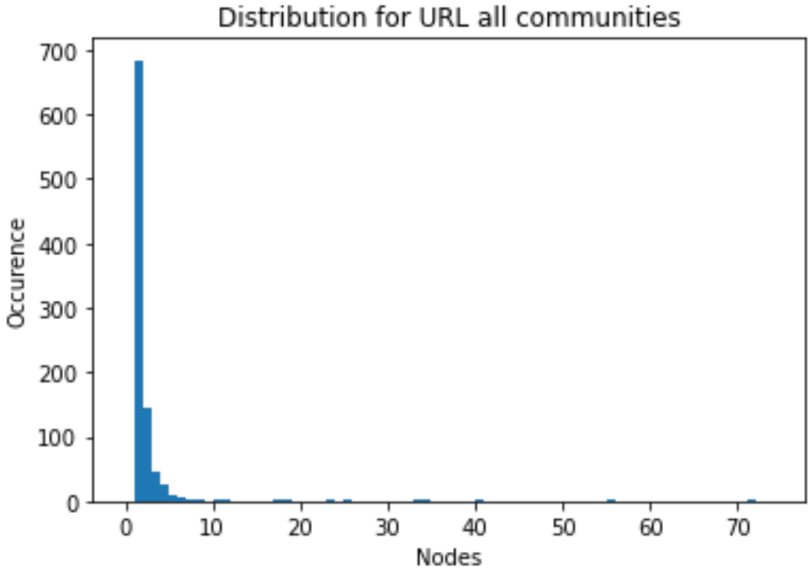


Figure 49: Distribution for nodes range 0 to 75

Furthermore, we note that our biggest community contains 17,078 nodes. Since there seems to be a huge difference between communities, we decided investigate which communities are the biggest. After exploring our data we decided to gather a group of communities having more than 100 nodes. We can see the results in table 12. In fact, here we are left with only 6 communities, meaning that there are 921 communities containing less than 100 nodes.

Community index	Community size
6	10491
233	2704
4	3375
7	17078
111	1234
11	2089

Table 11: Communities with 100 nodes and more URLs

At this point, we thought interesting to visualize the six communities, and since there are only a few groups, it will be easier to differentiate them. The Louvain-python package does have layouts to help us visualize our graph, however, in this case, it doesn't separate the communities properly, therefore we decided to use Gephi, a software specialised in graph visualization. As stated in chapter 2, Gephi can't work with too many nodes; 3000 is the approximate maximum. Therefore, to use Gephi, we have to remove the nodes that don't belong to communities with 100 nodes and more from the original graph, it is easier than creating a whole new graph. Our original graph contains 38,650 nodes, after removing the communities with 100 nodes and less, we are left with a graph containing 36,971. Since, this number is far higher than what we are allowed, we must take a percentage of each community, this will allow us to preserve the partition distribution. Technically, what we do, is remove close to 92% of nodes per community. Afterwards, we are left with a graph of containing 2841 nodes. In addition to this, we need to add the community index to the graph as an attribute to the node and save the graph. We can now open a project in Gephi and use the Atlas 2 template. After attributing a color to each community, we can visualize our graph. In image 55 we can see the six communities partitions and their occurrence.We note that nearly 75% of the output is made of two main communities.

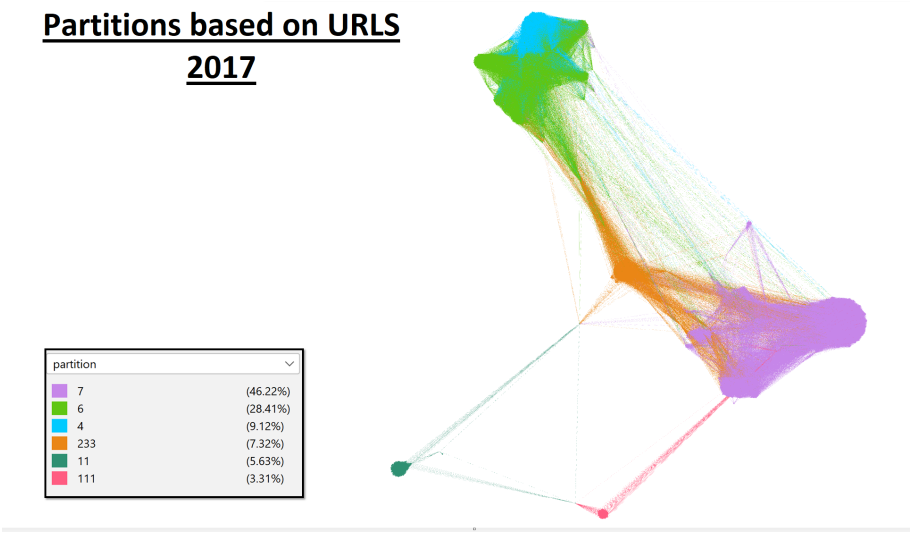


Figure 50: partitions graph based on urls 2017

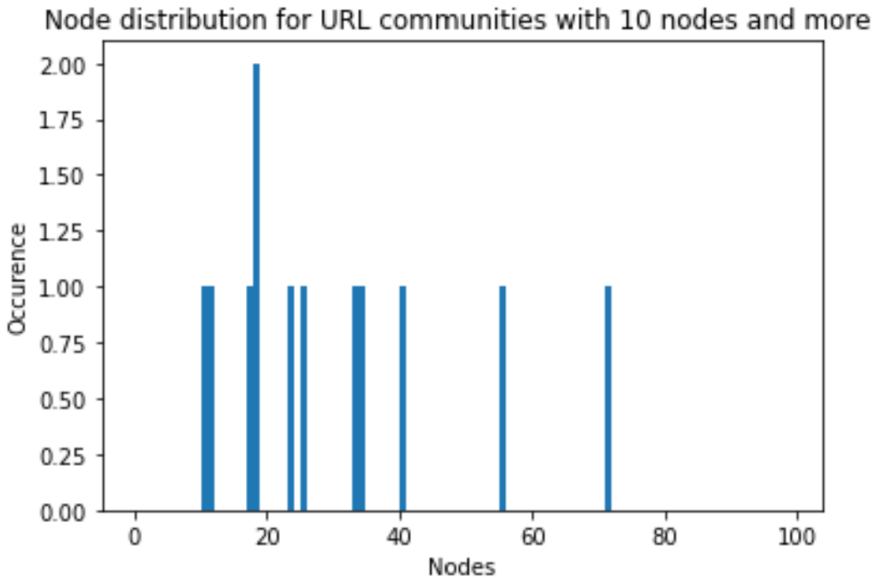


Figure 51: URL Distribution for 10 nodes and more

Our further analysis include smaller communities. We first look into communities containing 10 nodes or more. Since we already saw the distribution, we are not surprise to see that there are only 18 communities. Here in figure 52 we zoom into communities containing 10 nodes or more, we see that there is a disperse number of communities containing different number of nodes.

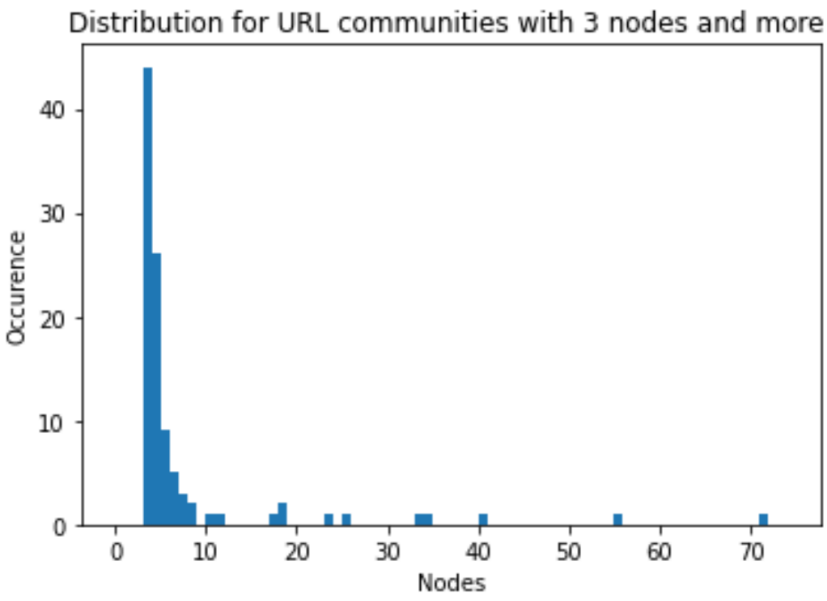


Figure 52: URL Distribution for 3 nodes and more

Secondly we look into communities containing 3 nodes or more. Here the number of communities reach 98, which is not that high of a number compared to the total number of communities in the partition. In image, 52 we zoom into communities with 3 nodes and more, and we note that most communities contain 3 to 5 nodes.

USERNAME Louvain analysis results

Here, we apply the same steps we followed in the previous section, therefore we start by producing our Graph G and run Louvain's algorithm to get a dendrogram as the main output. Here, the results shows only two levels: the first level has 2111 communities, while the best partition has 1397 communities. Here our biggest community contains 20,388 nodes. Since it will be near impossible to plot every single community, we create a graph that shows us the log distribution of all our nodes in our communities. Figure 53 is very similar to the one we saw for the URL analysis. Furthermore, if again, we zoom in into the communities having between 1 to 75, we note that 1158 of our communities consist of one node, which corresponds to 83% of our communities.

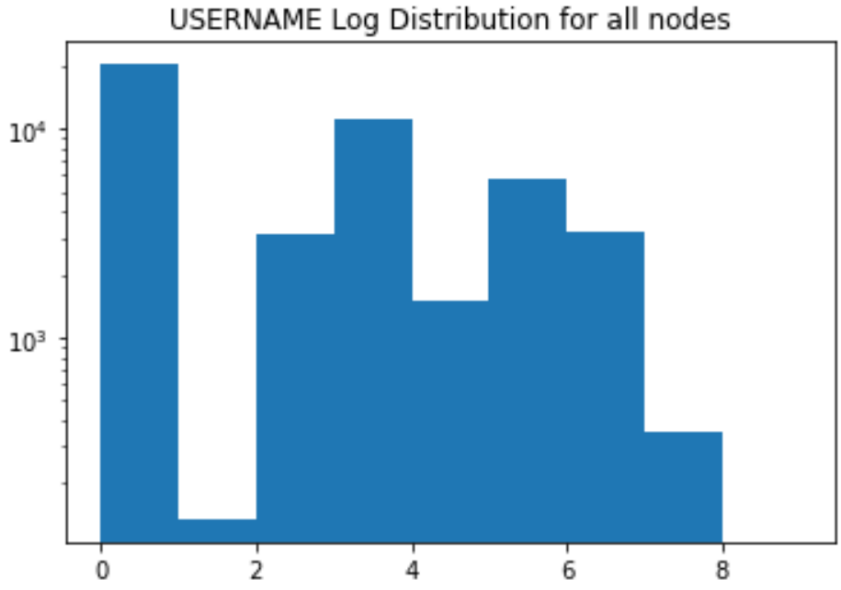


Figure 53: Log Distribution for USERNAME nodes

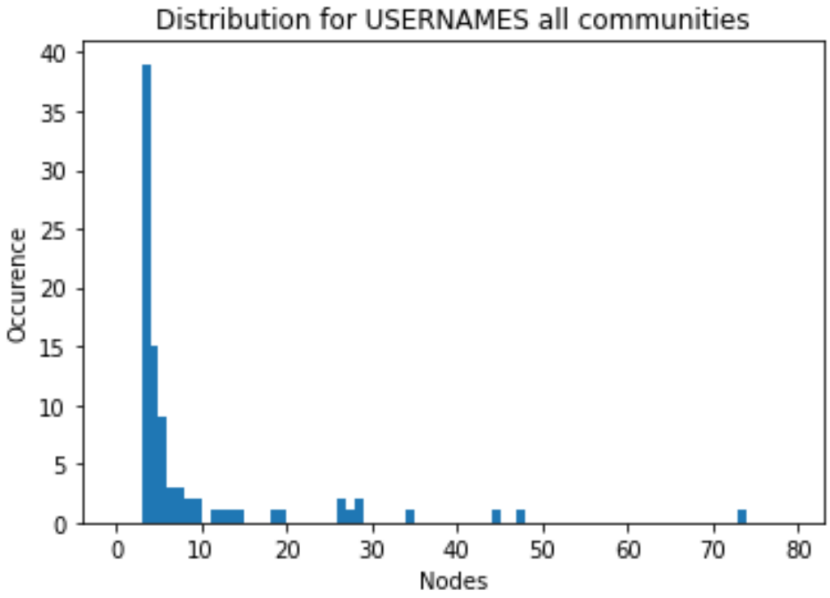


Figure 54: Distribution for USERNAME nodes range 1 to 75

Community index	Community size
24	10491
1	20388
47	430
373	4226
18	30044
89	1232
698	126
3	104
266	1110
492	2094

Table 12: Communities with 100 nodes and more

Just like we did for the Url analysis we created a new graph by removing the nodes that don't belong in partitions of 100 and more so we can use it to visualize our data. As mentioned before, since we are using Gephi can't work with more than 3000 nodes, so we have to reduce our graph. This is done by taking a percentage of each partition. We started with a graph containing 45,549 nodes, after removing the nodes allocated to paritions containing less than 100 nodes, we are left with 43,394 nodes. From here, we remove around 93% of the data per partition. We also remove all the nodes that are no longer part of those partitions from the graph. Now we are left with a graph containing 2,889 nodes.

Just as we did before, we need to add the partition number to the graph as an attribute to the node. Afterwards, we create a new project on Gephi and open the graph we just modified. Once again, we notice that we have two main communities, one is 47% of the graph and the other is nearly 25% of the graph.

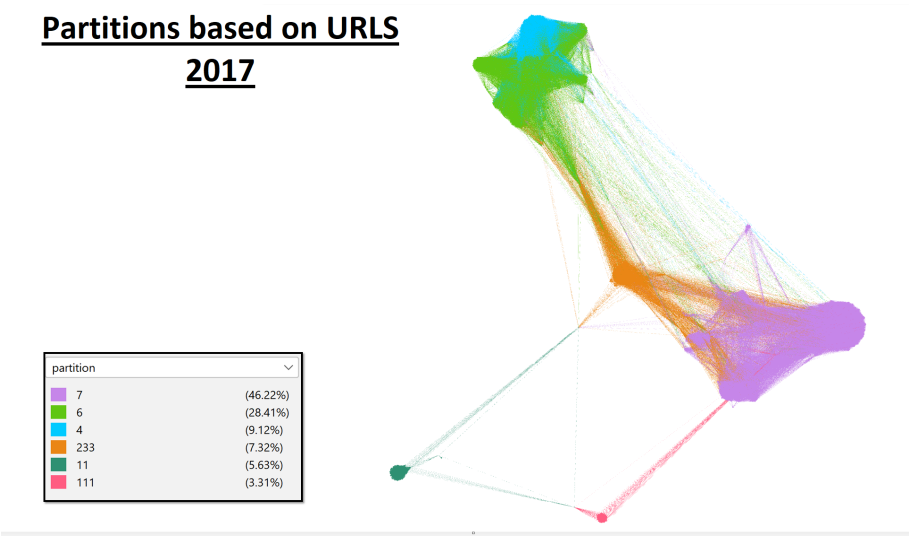


Figure 55: partitions graph based on urls 2017

Just like we did with our URL analysis, we examine our communities further by looking into communities that include less nodes. In fact, we look into communities having 10 nodes or more, the output shows us that we have 25 communities.In image 56 we zoom into this communities and we observe, just like in the URL analysis that our communities are more disperse. On the other hand, we have 98 communities with 3 nodes or more, and as we we can expect, most of the communities here do have 5 nodes or less, as we can see in figure 57.

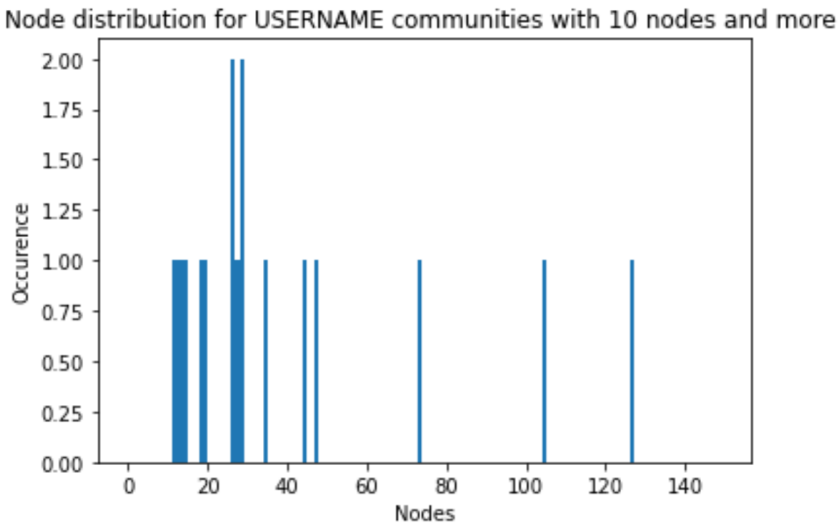


Figure 56: Distribution USERNAMES communities 10 and more

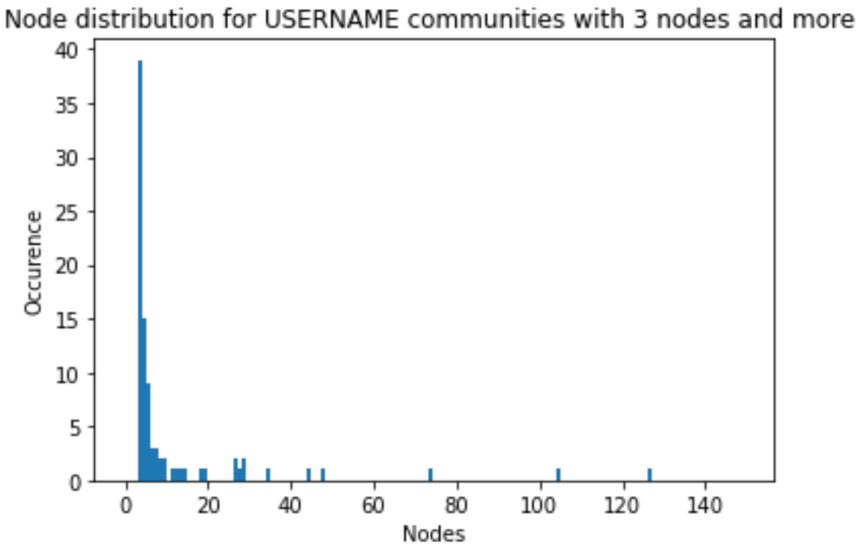


Figure 57: Distribution USERNAMES communities 3 and more

Comparison of Louvain URL and USERNAME

Here we want to compare our two previous analysis,namely the URL and the USERNAME Louvain community detection. We decided to apply two different methods of

comparison. First, we start by comparing both big communities, meaning both partitions containing communities of 100 nodes and more. This is done by iterating and comparing each group of node pertaining to a specific community in one feature analysis to the other. We are left with figure 58. The values in the matrix represent the total intersected nodes between both the URL and the USERNAME community. We added a simple map heat color code, only to show to show the extend of the overlapping when there is one. We added a percentage besides the values we deemed to be important intersects. This percentage is calculated by dividing the URL community size by the total number of nodes. Here we realize that for each URL community, we have an overlap of 85% and more with USERNAMES communities. We even have communities with 99% overlap. These results encouraged us to include the smaller communities to our analysis.

Confusion Matrix Best Partition 2017								
			URLS					
		Community Size	3375	10491	17078	2704	2089	1234
	Community Size	Community Number	1	2	3	4	5	6
USENAMES	20388	1	2	198 (4.4%)	16205 (95%)	233 (8.6%)	1	12
	2094	2	0	0	0	2	2081 (99.6%)	0
	104	3	0	58	33	0	0	0
	1110	4	1	703 (7%)	8	2	0	0
	430	5	0	4	238	0	0	0
	3004	6	0	4		2416 (89%)	0	0
	4226	7	3256 (96.5%)	463	5	1	0	0
	10680	8	108	8920 (85%)	35	10	0	0
	1232	9	0	0	14	0	0	1216 (99%)
	126	10	0	0	1	0	7	0

Figure 58: confusionMatrix2017

In short, we applied the normalized_mutual_info_score which was previously explained, to our both our partitions containing smaller communities. Note that we worked with communities of 3 nodes and more and 10 nodes and more. We discarded the rest simply because our further work won't include these smaller clusters. Since th normalized_mutual_info function compares two arrays that are ordered in the same manner, we mapped and united all our results in the same dataframe, in other words we mapped and ordered or data by merging dataframes together. For communities of 10 nodes our output was 0.77 while it was 0.72 for communities of 3 nodes and more. This was a satisfying result considering that it is in theory more difficult to overlap the smaller communities.

Our results gave us the confidence to pursue this lockstep venture by adding an additional observation: Infoshield. Furthermore, since our communities containing 3 nodes and more gave us a high score, we decided to continue or evaluation based on these communities. In this step, we also looked for the number of intersections between these two communities and find an overall of 44 intersections.

Comparison of Louvain with Infoshield

In this step we find and analyse the groups made of intersects between Louvain Communities and Infoshield. As discussed before, infoshield clusters the data based on its text. Consequently, each tweet has an Infoshield cluster number. On the other hand, each node or each author id, as a corresponding URL Louvain community and an USERNAME Louvain community. To compare them, we run Infoshield on dataframes that already have Louvain communities' assignments. Since there are two Louvain community analysis, we decided to run Louvain on both, we conclude that both Infoshield results will be very similar. Therefore, our choice will depend on further analysis of the data, however, both can be run. Note that we could have merged the communities results first and then proceed to apply the Infoshield method, but, in our case, a choice had to be made because of the way we chose to run the experiment (this is what we ended up doing in the paper).

Before starting the comparison of our three sets of groups, we need to verify if this step is necessary and feasible, therefore we apply the normalized_mutual_info_score on Communities and Infoshield. The comparison between the URL community versus the Infoshield community gives us a correlation of 0.18 while the USERNAME community versus the Infoshield community resulted in a correlation of 0.20. These correlations are not as strong as the ones gathered from solely the Communities comparison, however for the sake of testing, we also run these analysis on 'small' communities and 'big' communities separately. Here we consider that a big community contains 100 nodes and more while a small communities contains between 99 nodes and 3 nodes. The output of produced by the URL big communities and their associated Infoshield clusters is 0.17,

but the score produced by the smaller communities and the Infoshield clusters is slightly higher reaching 0.24. The results produced by the USERNAMES analysis are 0.19 and 0.47 respectively. As we note, the correlation between smaller communities and the Infoshield clusters is nearly 0.5, which is very high. Despite the lower correlation in respect to the URLs, we still believe it will be possible to find intersects between our different groups, however this is already giving us an indication as to say that Infoshield removes potentially interesting groups from our pool of intersections.

In fact, after running our simple algorithm presented in the previous chapter. Depending of which set of Infoshield cluster is chosen, we get 14 or 16 groups of intersects. Note that our code only takes in consideration groups that have at least 40% of their nodes in common (we are again using the Partial Intersection aglorithm found in the paper), that way we discard uninteresting groups from the start. Note that the analysis done with USERNAME Infoshield as two EXTRA groups, but all the other groups are the same as the ones found in the analysis using URL Infoshield. the The reason why it contains more groups of intersects is simply because the Infoshield clusters derived from USERNAMES showed more correlation with the USERNAME communities.

Figures 59 and 60 are the two outputs we have, each line gives the iteration number for the position of the group in the list it is iterating, for example, here we start with the URL community, so we get the iteration number then the community size, this is follow by the same information for the USERNAME community and Infoshield cluster groups. Underneath each line, we get the number of intersects. Note that sometimes this number is very small because we have a very small cluster. No matter the size of these groups of authors, they still might have multiple tweets which may contain multiple URLS or USERNAMES, therefore, they are still of interest for us.

```
intersecion between community # 1 4  and between community # 12 5 intersecion between cluster # 114 3
3
intersecion between community # 8 3  and between community # 96 4 intersecion between cluster # 122 3
3
intersecion between community # 24 18  and between community # 16 18 intersecion between cluster # 23 18
18
intersecion between community # 31 34  and between community # 21 34 intersecion between cluster # 97 33
33
intersecion between community # 33 4  and between community # 423 5 intersecion between cluster # 99 2
2
intersecion between community # 34 44  and between community # 31 33 intersecion between cluster # 84 32
31
intersecion between community # 40 4  and between community # 32 4 intersecion between cluster # 153 2
2
intersecion between community # 55 3  and between community # 52 3 intersecion between cluster # 146 2
2
intersecion between community # 64 7  and between community # 549 7 intersecion between cluster # 170 5
5
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 10 25
25
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 62 22
22
intersecion between community # 73 3  and between community # 292 4 intersecion between cluster # 112 2
2
intersecion between community # 87 3  and between community # 343 4 intersecion between cluster # 124 2
2
intersecion between community # 88 5  and between community # 346 5 intersecion between cluster # 132 5
5
total number of intersections: 14
```

Figure 59: Intersect 2017 based on URL Infoshield

```
intersecion between community # 1 4  and between community # 12 5 intersecion between cluster # 54 3
3
intersecion between community # 8 3  and between community # 96 4 intersecion between cluster # 57 3
3
intersecion between community # 24 18  and between community # 16 18 intersecion between cluster # 90 18
18
intersecion between community # 31 34  and between community # 21 34 intersecion between cluster # 61 33
33
intersecion between community # 33 4  and between community # 423 5 intersecion between cluster # 29 2
2
intersecion between community # 34 44  and between community # 31 33 intersecion between cluster # 13 31
31
intersecion between community # 40 4  and between community # 32 4 intersecion between cluster # 103 2
2
intersecion between community # 41 8  and between community # 812 8 intersecion between cluster # 49 13
8
intersecion between community # 55 3  and between community # 52 3 intersecion between cluster # 86 2
2
intersecion between community # 64 7  and between community # 549 7 intersecion between cluster # 94 5
5
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 39 22
22
intersecion between community # 71 47  and between community # 66 55 intersecion between cluster # 96 25
25
intersecion between community # 73 3  and between community # 292 4 intersecion between cluster # 7 2
2
intersecion between community # 82 3  and between community # 691 4 intersecion between cluster # 23 3
2
intersecion between community # 87 3  and between community # 343 4 intersecion between cluster # 92 2
2
intersecion between community # 88 5  and between community # 346 5 intersecion between cluster # 10 5
5
total number of intersections: 16
```

Figure 60: Intersect 2017 based on USERNAME Infoshield

After getting our list of groups of intersects, we have to inspect each one and count how many unique URLS and unique USERNAMES they contain. For this purpose, we loop and filter our merged dataframe by each of the groups of intercept found. This process outputs two arrays. We filter our intersections by keeping the groups containing more than unique URL and USERNAMe.The only group filtered that we found contained 2 unique URLs and 3 unique USERNAMES. Note that seven out of the 16 intersections contained 3 or more USERNAMES but had only one unique URL. These are overall poor results. Based on this, we decided to do a new analysis without Infoshield, furthermore, our new analysis would be also done on communities containing 2 nodes (new analysis for the paper - we also ended up collecting new data).