**Marie Sklodowska Curie,
Research and Innovation Staff
Exchange (RISE)**

European Commission | Horizon 2020
European Union funding
for Research & Innovation

ENhancing seCurity and privAcy in the Social wEb: a user-centered approach for the protection of minors

ENCASE
ENhancing seCurity and privAcy in the Social wEb

WP4 – User Profiling for Detection and Prediction of Malicious Online Behavior

Deliverable D4.2 "Software libraries built on Graphos.ml using data mining for the detection of aggressive or distressed behaviors in OSN"

Editor(s):	Athena Vakali (AUTH), Vaia Moustaka (AUTH)
Author(s):	Michael Sirivianos (CUT), Peter Papagiannis (CUT), Antonis Papasavva (CUT), Savvas Zannettou (CUT), Konstantinos Papadamou (CUT), Charalampos Partaourides (CUT), Sotirios Chatzis (CUT), Emiliano De Cristofaro (UCL), Juan Echeverria Guzman (UCL), Catherine Holloway (UCL), Athena Vakali (AUTH), Antigoni Maria Founta (AUTH), Emmanouil Rigas (AUTH), Despoina Chatzakou (AUTH), Lia Terzidou (AUTH), Irene Papagiannopoulou (AUTH), Vaia Moustaka (AUTH), Rafael Constantinou (CYRIC), Ioannis Agrotis (CyRIC), Demetris Antoniades (CyRIC),
Dissemination Level:	Public
Nature:	Report
Version:	0.9

ENCASE Project Profile

Contract Number	691025
Acronym	ENCASE
Title	ENhancing seCurity and privacy in the Social wEb: a user-centered approach for the protection of minors
Start Date	Jan 1st, 2016
Duration	48 Months

Partners

	Cyprus University of Technology	Cyprus
	University College London	United Kingdom
	Aristotle University	Greece
	Universita Degli Studi, Roma Tre	Italy
	Telefonica Investigacion Y Desarrollo SA	Spain
	Cyprus Research and Innovation Center, Ltd	Cyprus
	SignalGenerix Ltd	Cyprus
	LSTech	United Kingdom

Document History

AUTHORS

(CUT)	Michael Sirivianos, Peter Papagiannis, Antonis Papasavva, Savvas Zannettou, Konstantinos Papadamou, Charalampos Partaourides, Sotirios Chatzis
(UCL)	Emiliano De Cristofaro, Juan Echeverria Guzman, Catherine Holloway
(AUTH)	Antigoni Maria Founta, Athena Vakali, Despoina Chatzakou, Emmanouil Rigas, Lia Terzidou, Irene Papagiannopoulou, Vaia Moustaka
(CYRIC)	Rafael Constantinou, Ioannis Agrotis, Demetris Antoniades

VERSIONS

Version	Date	Author	Remarks
0.1	03.07.2018	AUTH	Initial Table of Contents (TOC)
0.2	08.10.2018	AUTH	Complete TOC
0.3	10.11.2018	All Authors	Contributions from Partners Received
0.4	23.12.2018	AUTH	Initial Draft of the Report
0.5	03.12.2018	AUTH + CUT	Revision and Restructure
0.6	04.12.2018	CUT	Revision and Addition of Introduction
0.7	08.12.2018	AUTH+CUT	Revision
0.8	12.12.2018	AUTH+CUT	Revision and Restructure
0.9	13.12.2018	CUT	Final Version

Executive Summary

One of the main objectives of the ENCASE project is to develop a browser add-on and its corresponding machine learning algorithms for the detection of malicious and problematic behavior such as cyberbullying, hate speech, aggressive behavior, distressed behavior, and sexual grooming.

This document describes the efforts and algorithms developed to identify and quantify the various types of online abuse. It also explains the research conducted and the methodologies studied to detect hateful content, raid, abuse, sexual grooming and bullying.

This deliverable also provides a brief description of the completed projects carried out towards automatically detecting malicious behavior in the context of tasks T4.1 - "User profiling to detect and prevent malicious and criminal activities" and T4.2 - "Sentiment and affective analysis on individual and collective basis", which were listed as "ongoing" and were included in deliverable D4.1 – "Development of automated techniques to detect early indications of malicious behavior of social network users". Importantly, the work and efforts that took place during Task 4.3 - "OSN malicious users time-dependent detection" is thoroughly presented herein.

Task 4.3 aims at modeling of time-dependent interactions and activities of social network users, and the application of graph mining and text processing methodologies to detect latent patterns of activity by users/entities and their interactions. Advanced data mining and analytics techniques were developed in order to leverage the OSN users' concurrent activities that indicate behavioral variations and spikes with emphasis on advancing the state of the art on anomaly detection in OSN.

Table of Contents

List of Figures

List of Tables

1. Introduction

Unfortunately, bullying is a big part of the life of every youngster in our days. Recent study announced that over 3.2 million students are victims of bullying each year and approximately 160,000 teens skip school every day because of bullying. Notably, 17% of American students report being bullied 2 to 3 times a month or more within a school semester[1].

The worst part of bullying is when the event takes even bigger extent and reaches the house of the youngsters. Today, bullies use the internet and Online Social Networks (OSN) to spread their hatred and loathing. Cyberbullying, cyber aggression, hate speech, racism and sexual grooming and the lack of cyber safety are serious and widespread issues affecting increasingly more Internet users.

This document attempts to address the above mentioned problems. Researchers developed automated techniques with the aim to detect early indications of malicious behavior, how malicious behavior affects the feelings of young users, and how to better protect minors from online radicalization. Specifically:

a) In Section 2, we go through the research done for Behavioral Pattern Identification and how we can identify different predator classes based on their behavior;

b) In Section 3, we try to identify available datasets including chat conversations between what can be considered benign users, in a friendly context;

c) In Section 4, we focus on identifying, extracting and cleansing of group conversations with the purpose of extracting bidirectional friendly conversation datasets between two group participants;

d) In Section 5, we explain how we tried discovering multiple patterns indicative of predatory (sexual predators or cyberbullying) behavior over time by analyzing OSN user interactions;

e) In Section 6, we put all the above in practice and we aim to detect cyberbullying behavior against minors. Our approach is through the emotional state of the minor. If a minor feels angry/sad/frustrated during a chat conversation, then this might be an early indication cyberbulllying;

f) In Section 7, we present a large-scale, quantitative study of online antisemitism;

g) Last, in Section 8, we provide an assessment of the popularity and diversity of memes and how they are extremely common in fringe Web communities.

[1] https://www.dosomething.org/us/facts/11-facts-about-bullying

2. Behavioral Pattern Identification

2.1. Project Description and Motivation

The purpose of this work was the development of Python scripts and equivalent APIs for the visualization and analysis of the Perverted Justice dataset. Specifically, the temporal analysis of the Perverted Justice dataset for identification of different predator classes based on their behavior was conducted, while the visualization of the statistics which was emerged from the Perverted Justice overall analysis for all cases and all sessions of the dataset was carried out.

2.2. Perverted Justice Dataset Correction and Temporal Analysis

During the processing of the Perverted Justice dataset, some inconsistencies in it were identified, which led to its correction and enrichment. The main issues were the following:

i. In some cases the username of the predator and/or victim would change within the same case. This lead to wrong value in the "PV" field (i.e., this field describes a participant as predator (P) or victim (V)).
ii. Wrong values for the "Date" existed. In some cases, the time changed, but the date remained unchanged (e.g., a message sent at 23:59 and another one sent at 00:01 had the same date).

The *first problem* was solved following the next steps: i) the first message of the predator and the victim are identified, ii) to identify the cases with possible errors the number of messages from predators and victims were calculated. A Python script was developed for this purpose. The ones with big difference between the two values were examined for possible errors, iii) the different usernames for the predator and victim were collected manually, iv) all different usernames identified for predators and victims were replaced by the ones found in step 1, and v) the values for "PV" were updated according to the username.

The *second problem* was solved following the next steps: i) the values for the "Date" fields of all cases were checked, ii) the date was transformed into a number of seconds, iii) the difference between this value for all pairs of chatlogs was calculated, iv) If this value was negative, this means that the date has not changed, and v) In this case, the value for the day and possibly the month was updated.

Once the correction of the dataset was completed, the next step was the enhancement of the *"Session"* field. A session can be defined as a series of messages exchanged within x seconds between each other. In our case, the values that were selected for x were 30 sec, 60 sec, 5 min and 30 min. Given that the aim of this task is the time-related analysis of the dataset, the addition of the sessions was crucial for the analysis undertaken in the following steps. Code-wise, a Python script was developed that was reading each case from the database and each chatlog of the case and was adding the session numbering based on the time that passed since the previous message from the current one. The equivalent value was added in the Session field which is of type Array. An example session counting can be seen in Table 1.

Table 1. Example session counting

Message time	30 sec	60 sec	5 min	30 min
10:01:05	1	1	1	1
10:01:15	1	1	1	1
10:01:47	2	1	1	1
10:04:34	3	2	1	1
10:31:01	4	3	2	1

2.3. APIs for Data Analysis

The APIs (Applications Programming Interfaces) that were developed for the visualization and the analysis of the data are presented in this section. Note that the example charts in this section are based on the Salsakewl case and a 5min session.

MessageCountingCall

MessageCoutingCall gets as input the name of a case and returns two two-dimensional arrays. The first contains the number of messages sent by the predator at each time and minute and the second is the equivalent for the victim (i.e., the arrays are of size 24X60). This API is useful to capture the number of messages that are being exchanged throughout the 24 hours of a day. This API can be used to calculate the total number of messages per hour (see Figure 1), as well as the number of messages for pair of hour and minute (see Figure 2). In the first case, we can have an indication of how active each participant is during the course of a day. For example, in the Salsakewl case, a lot of activity takes place in late evening and early morning times and very little during day time. In the second case, we can have an indication of how active each participant is during the course of a day and for each minute of each hour. In the Salsakewl case, the majority of the messages are exchanged in late evening and early morning hours and the messages are relatively evenly distributed within each of these hours.

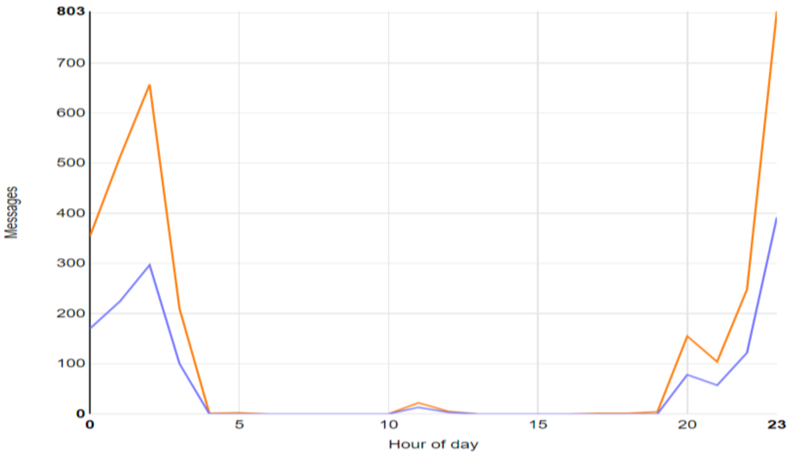


Figure 1. Messages per hour

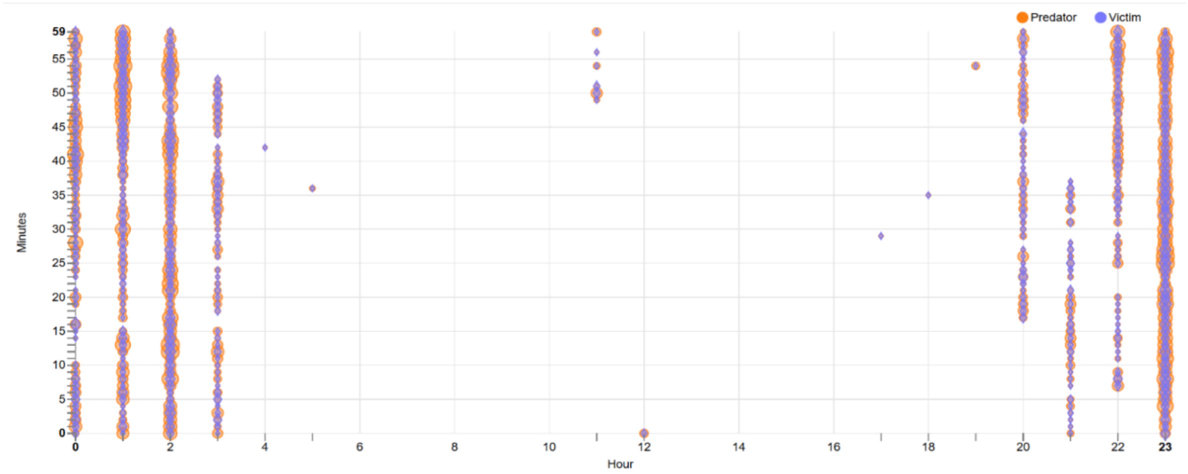


Figure 2. Messages per hour and minute

MessageCountingHourlyCall

MessageCountingHourlyCall gets as input the name of a case and returns a two-dimensional array containing the sum of messages for each second of an hour and each minute for all hours (i.e., the participants to be bots. In the Salsakewl case, no such pattern can be observed (see Figure 3).

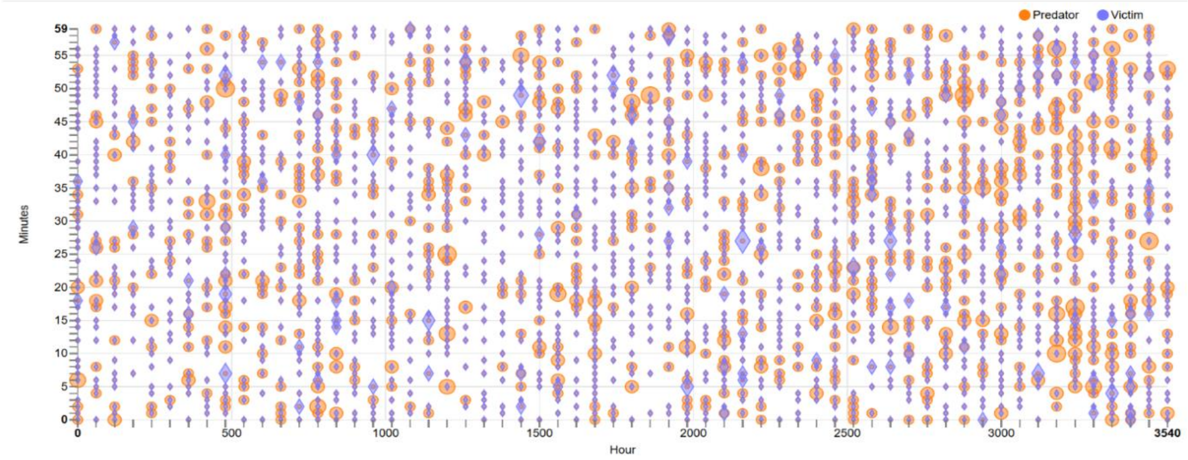


Figure 3. Messages per second and minute of hour

TotalMessagesPerSessionCall

TotalMessagesPerSessionCall takes as input the name of a case and the option for session window and returns the number of messages per session for the predators and the victims. These data give an indication of how the number of messages exchanged varies over time, as well as an indication of the density of the messages (see Figure 4). Moreover, they can show how the number of messages varies with the duration of the session. For example, in the Salsakewl case, the number of messages increases with the duration of the session.

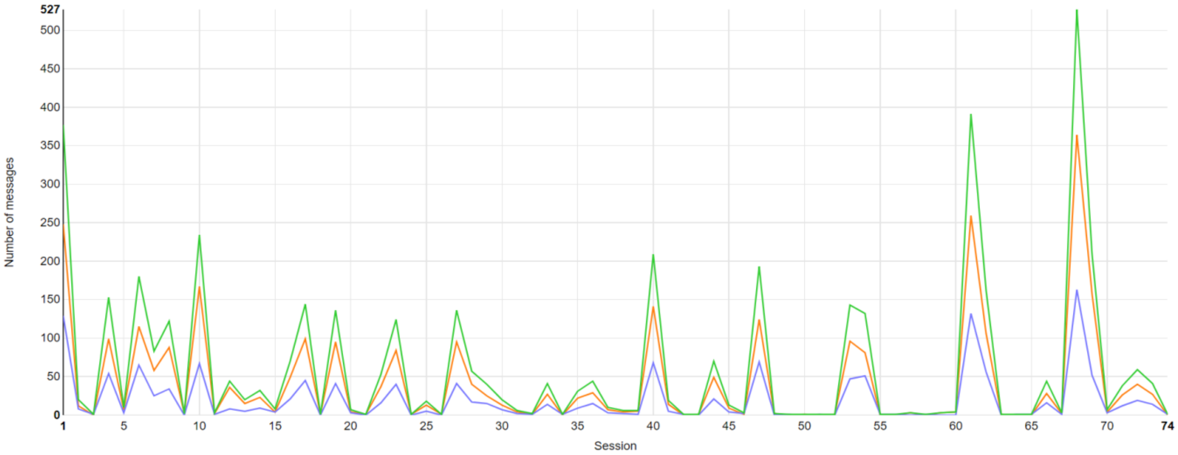


Figure 4. Messages per session

TotalMessagesPerDayCall

TotalMessagesPerDayCall gets as input the case name and returns an array with the number of messages for every day and for the duration of the case. These data can give an indication of the activity throughout the case period. In the Salsakewl case, an uneven distribution of messages can be observed (see Figure 5).

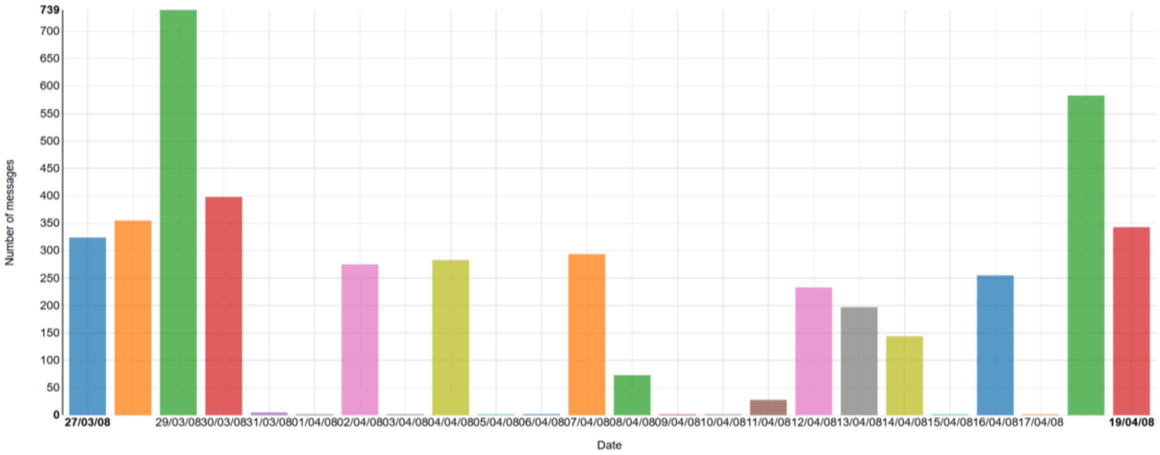


Figure 5. Messages per day

SessionInitiateCall

SessionInitiateCall takes as input the name of a case and the option for session window and returns two arrays with length equal to the number of sessions to show which sessions are initiated by predators or victims respectively. These data can give an indication of how willing each participant is to initiate a conversation with the predator, as well as an indication of the duration of the conversation initiated by predators and victims (see Figure 6). For example, in the Salsakewl case, when the session duration increases, the percentage of sessions initiated by the predator increases. This means, that the predator initiates conversations with longer duration and with higher density of message exchanging.

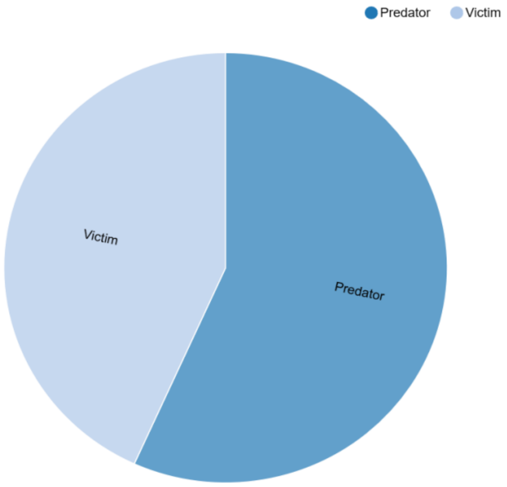


Figure 6. Session initiated by predator/victim

PredatorVictimReciprocalMessagesPerSessionCall

PredatorVictimReciprocalMessagesPerSessionCall calculates the number of messages for predator and victim and the sessions that are reciprocal (i.e., the sessions where only the predator is posting messages and the victim sends no reply). It takes as input the name of a case and the option for the session window and returns three arrays with the equivalent values. Regarding the data on reciprocal sessions, they can give an indication of how responsive the victim is to the predator’s messages and how this varies with the duration of the session. In the Salsakewl case, the percentage has small variations with the duration of the session. However, no trend can be identified, and no conclusion can be drawn (see Figure 7).

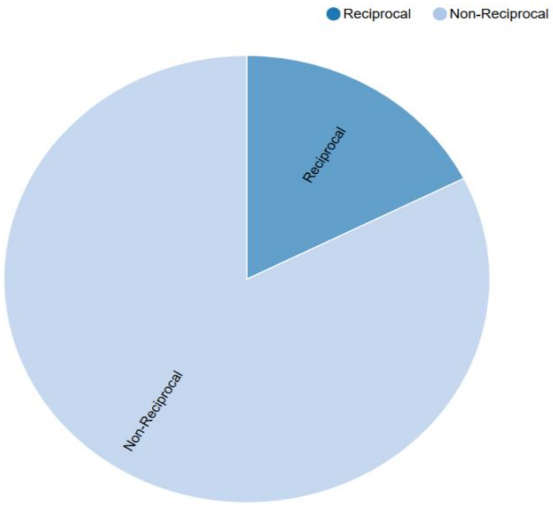


Figure 7. Sessions reciprocal

TimeToReplyCall

TimeToReplyCall takes as input the name of a case and the option for session and returns the average time that the predator takes to reply to the victim and the victim to the predator within each session. These data give an indication of how eager the predator and the victim are to reply to the messages. Note that in some cases/chatlogs the seconds in the timestamp were not available.

For this reason, some zero values exist. Also note that the duration of the session sets the upper limit in the average time to reply. In the Salsakewl case, the predator has a shorter time to reply compared to the victim (see Figure 8).

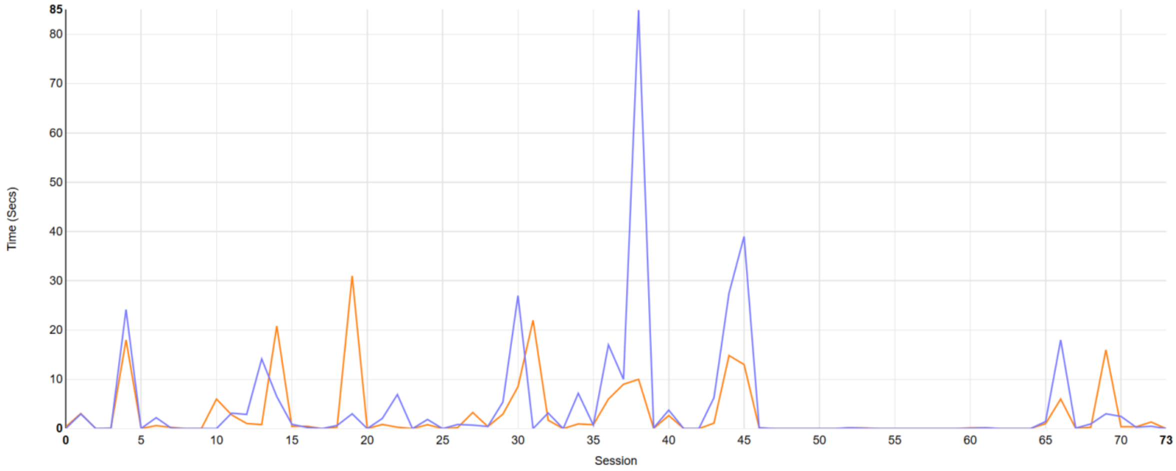


Figure 8. Response time per session

CDFCalculationCall

CDFCalculationCall calculates the CDF for each case. An array with number of messages and percentage of sessions having at most this number of messages. In the CDF graph, the x-axis represents the number of messages and the y-axis the percentage of cases that have at most the number of messages in the equivalent x-axis value. It is expected the graph to reach the 100% value sooner as the duration of a session increases. This is the case in the Salsakewl case (see Figure 9).

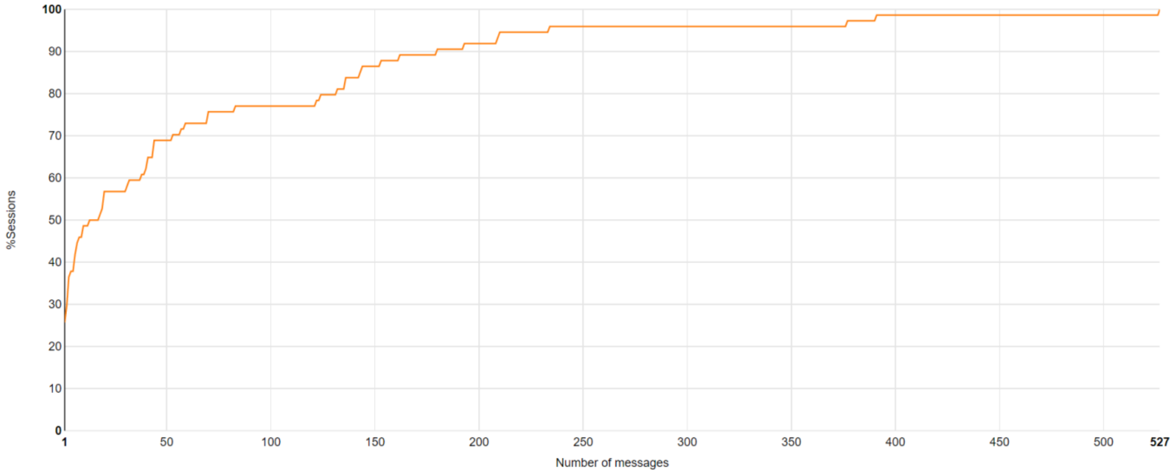


Figure 9. CDF for sessions and messages

MessagesPerTimeWindowCall

MessagesPerTimeWindowCall takes as input the name of a case and the start and end time and returns the number of messages for predators and victims for the given time window.

SessionsPerDayWindowCall
SessionsPerDayWindowCall takes as input the name of a case and two dates and returns the number of sessions between these two dates.

2.4. Investigation of Reciprocal and Non-Reciprocal Sessions Between Users

The temporal analysis on the Perverted Justice dataset was extended with the purpose of identifying of different predator classes based on their behavior focusing on the data analysis for all cases investigating reciprocal and non-reciprocal sessions in order to provide a visual overview of the whole dataset, using statistical diagrams (i.e., Scatter Plots, CDF, Box Plots). The developed Python scripts for the analysis and the visualization of the extracted statistics for the Perverted Justice dataset are presented below.

Note that the example diagrams in this section concern statistics for all cases. The sessions are identified based on a time window between the two consecutive messages from the two participants. We define as a reciprocal session one where both participating parties have posted messages. All other sessions where only one of the parties has sent one or more messages are considers non-reciprocal.

FindReciprocalStatsPerCase
FindReciprocalStatsPerCase gets as input: a list with all chats for a given case, the name of the case, the Id of the session type (i.e., session Id = 1 for 30 sec sessions), a list of dictionaries (*SessionsStats*) for storing the statistics for each session of the given case and returns: a dictionary named *case_stats* with the statistics for the given case and session type. More specific, the *case_stats* dictionary includes the following fields: *case (name of the case), sessionType*, *sessionsCount*, *SessionsReciprocal*, *SessionsNonReciprocal*, *SessionsReciprocalPercentage*, *SessionsNonReciprocalPercentage*, *CaseChats*, *PredMsgsCount*, *VictMsgsCount*.

The *SessionsStats* list variable is passed by reference and is used to capture detailed data for each session of the given case. To extract the statistics for each case and each session of the case, the method is called in a loop for each document/case of the database in the main program. The returned *SessionsStats list*, provides the needed data for the visualization of the statistics related to sessions (reciprocal/non-reciprocal) and the number of messages. The *SessionsStats* data was used to extract the diagrams related to sessions’ statistics. More specific, the case_stats list includes the following fields: *sessIndex* (gives a unique identifier to each session, *Case* name, *sessId*, *sessType*, *sessIsReciprocal* (true or false), *sessMsgsCount* (counts messages per session, *caseMsgsCount* (counts *total _chats* for the given case).

ScatterNonReciprocalAllCases
ScatterNonReciprocalAllCases gets as input the list containing the data for each case and plots a scatter plot. In detail, the scatter diagram visualizes the percentage of all non-reciprocal sessions for all cases (see Figure 10). We can observe that non-reciprocal sessions are presented in almost all cases with most of the cases ranging between 10-40% of all the sessions. However, the analysis of a session type with a larger time window, i.e. session type 3, shows that the percentage of non-reciprocal sessions widens dispersed ranging from 10%-80% (see Figure 11).

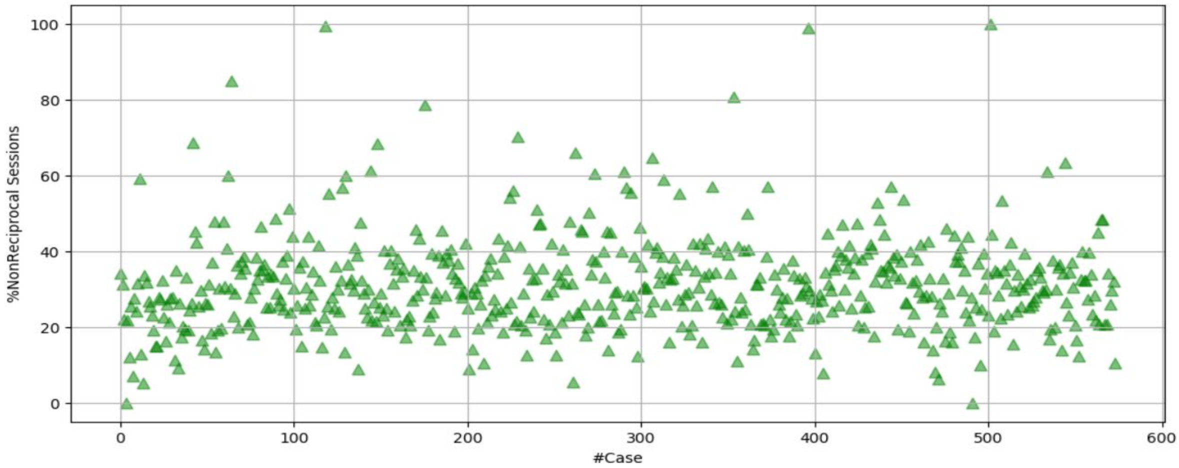


Figure 10. Percentage of Non-Reciprocal Sessions per Case | Session Type 1

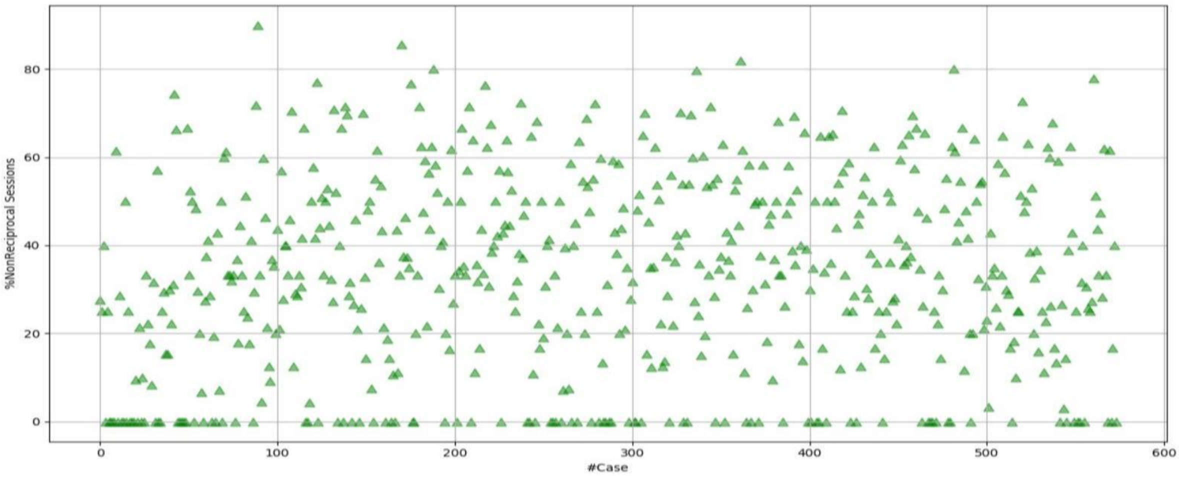


Figure 11. Percentage of Non-Reciprocal Sessions per Case | Session Type 3

CDFNonReciprocalAllCases

CDFNonReciprocalAllCases calculates the CDF for all cases. It gets as input the list containing the data for each case and plots a CDF [cumulative distribution function] diagram for the non-reciprocal sessions. In the CDF diagram, the x-axis represents the cumulative percentage of cases and the y-axis the cumulative percentage of Non-Reciprocal sessions corresponding to a specific value of percentage of the x-axis. The CDF diagram in Figure 13 visualizes the cumulative percentage of non-reciprocal sessions for all cases of the Perverted Justice dataset. We can observe that 90% of all cases correspond to about 50% of the non-reciprocal sessions for session type 1 (Figure 12). Similar is the Figure 14 that results from the analysis of sessions with a larger time window (Figure 13).

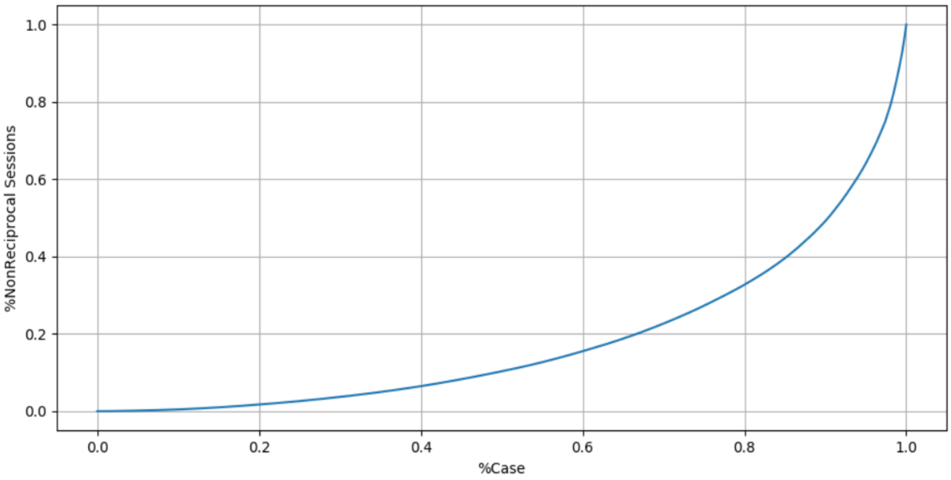


Figure 12. CDF of Non-Reciprocal Sessions for All Cases | Session Type: 1

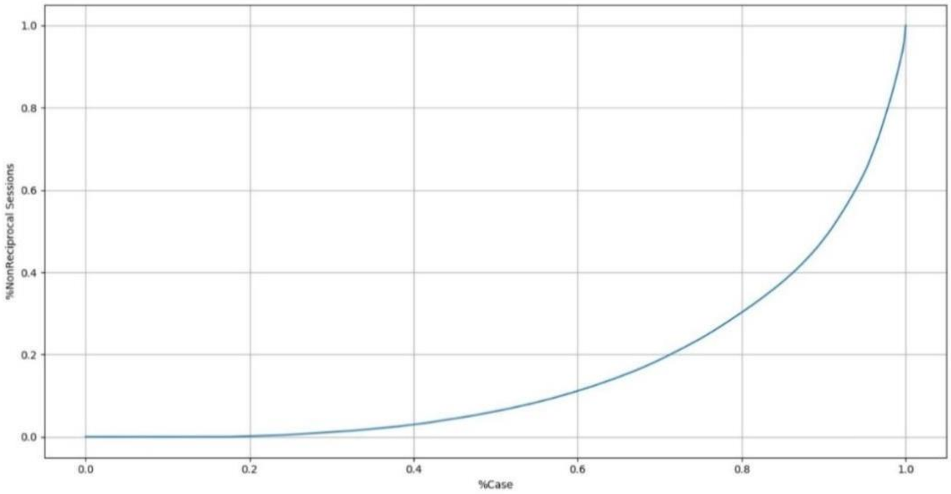


Figure 13. CDF of Non-Reciprocal Sessions for All Cases | Session Type: 3

Scatter Plots for reciprocal and non-reciprocal sessions for all cases

- **ScatterMsgsNonReciprocalAllSessions** gets as input a list containing data for each session of each case (data for sessions were stored in the SessionsStats list of dictionaries structure) and plots a scatter plot. In detail, the scatter diagram visualizes the number of messages for each non-reciprocal session for all cases (Figure 14(a)). We can observe that a low number of exchanging messages characterizes almost all non-reciprocal sessions, limited to less than 10 messages per session.

- **ScatterMsgsReciprocalAllSessions** gets as input a list containing data for each session of each case (data for sessions were stored in the SessionsStats list of dictionaries structure) and plots a scatter plot. In detail, the scatter diagram visualizes the number of messages for each reciprocal session for all cases (Figure 14 (b)). We can observe that the exchanging messages for most of the reciprocal sessions are presented in less than 100 messages per session and just few of them exceed the 100 messages.

- **ScatterMsgsAllSessions** gets as input a list containing data for each session of each case [data for sessions were stored in the SessionsStats list of dictionaries structure] and plots a scatter plot. In detail, the scatter diagram visualizes the number of messages for each session [reciprocal and non-reciprocal] for all cases (Figure 14 (c)). We can observe that the exchanging messages for all sessions are presented in less than 100 messages per session and just few of them exceed the 100 messages.

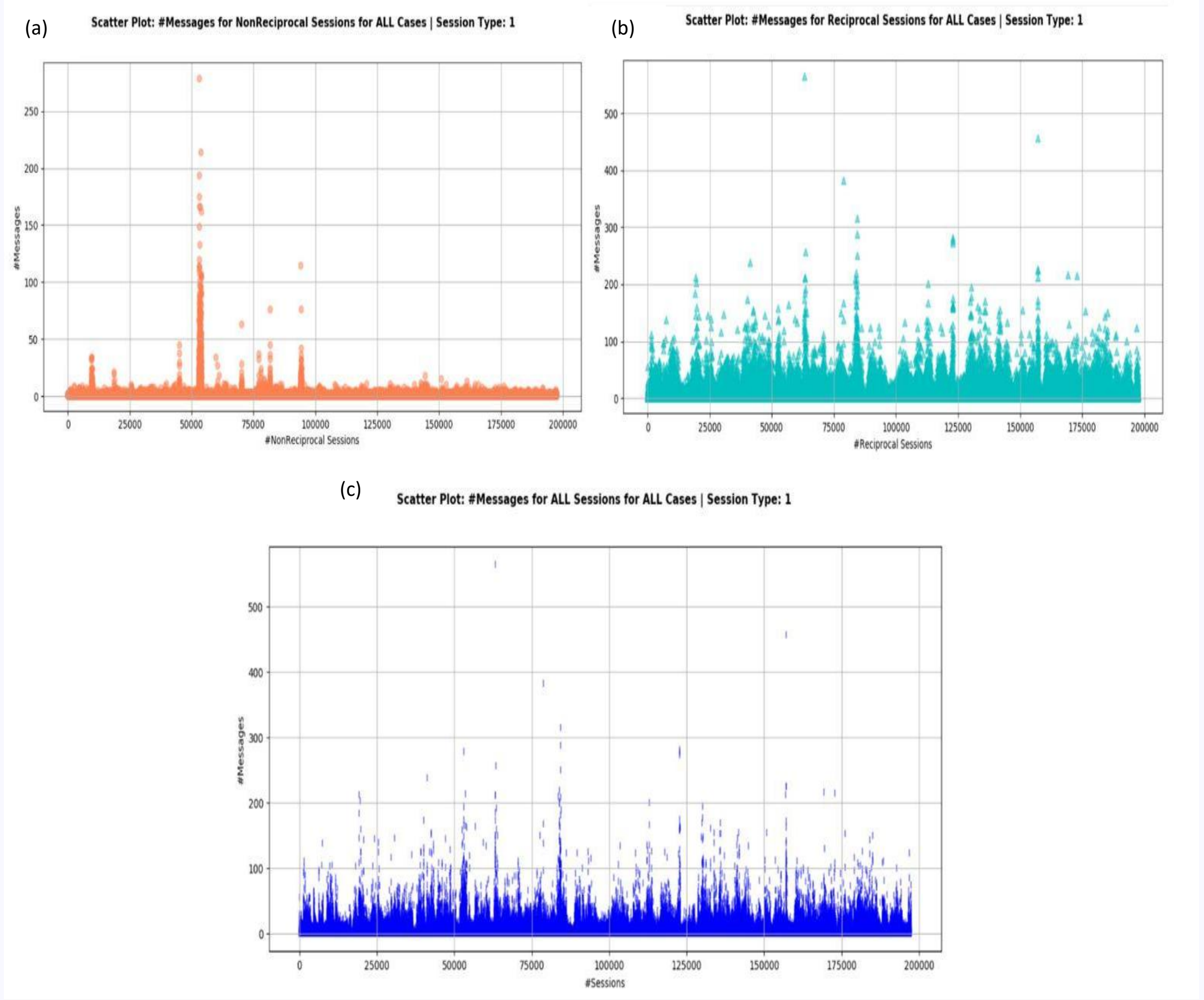


Figure 14. Messages of Non-Reciprocal, Reciprocal and all Sessions for All Cases | Session Type: 1

Furthermore, the increment of the time window between chats (i.e., session type 3) confirms the low number of exchanging messages between the victim and the predator, which is, for non-reciprocal sessions, presented in less than 100 messages per session for almost all the sessions (Figure 15 (a), (b), (c)). Regarding the reciprocal sessions we notice that for the longer time window of session type 3, the exchanging messages increase for most of the reciprocal sessions and are presented in less than 250 messages per session.

checkIfSessionIsReciprocal

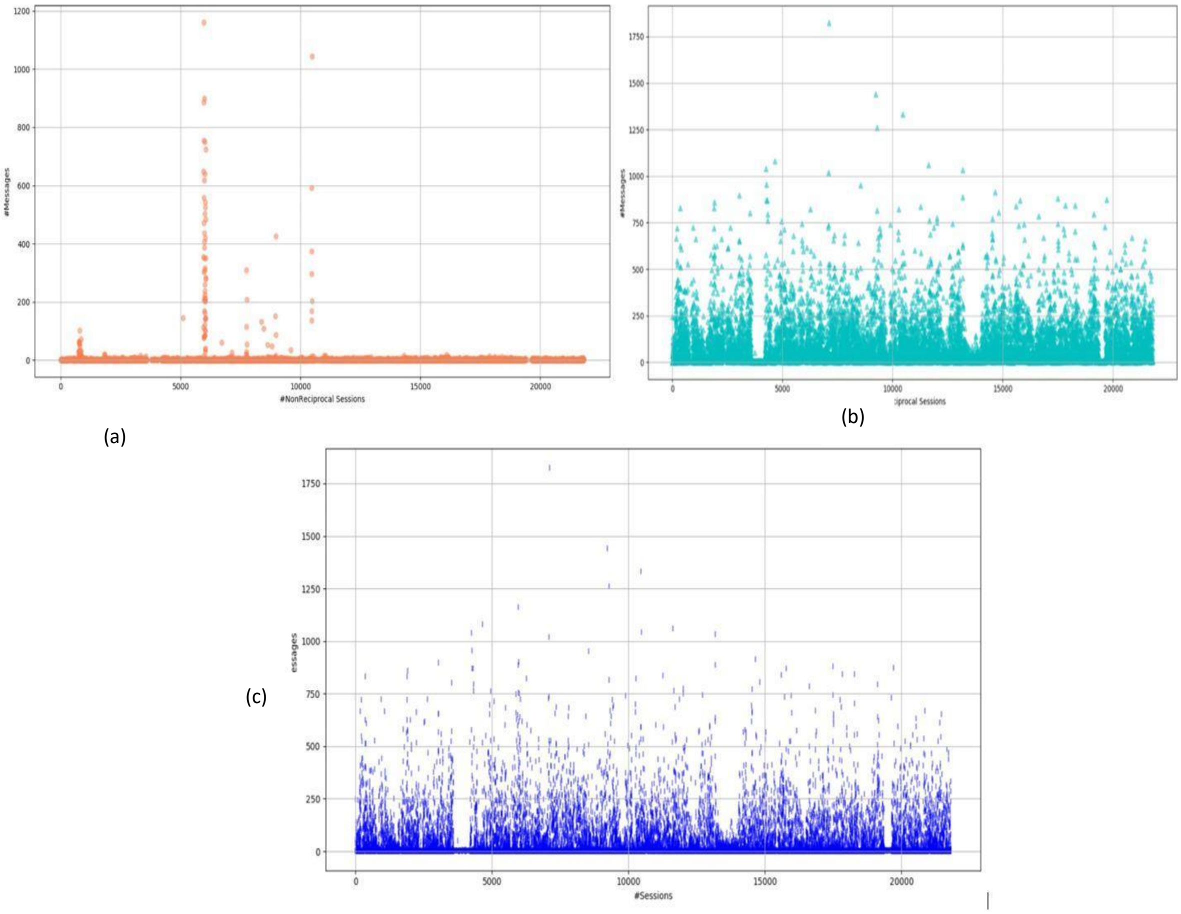
checkIfSessionIsReciprocal gets as input the total number of predator’s messages and the total number of the victim’s messages name of a case and returns a Boolean value that indicates if the session is reciprocal (returns True) or non-reciprocal (returns False).

percentage

percentage gets as input two numbers (*part* and *whole)*, calculates and returns a float number that represents the relation between *part* and *whole* expressed in percentage.

CaseStatsPrint

CaseStatsPrint prints the statistics for all cases of the Perverted Justice dataset. It gets as input a list of dictionaries, which contains useful statistical data for all cases and sessions (reciprocal and non-reciprocal). Although it was initially developed for testing and verifying the visually provided statistics based on these data, it is provided in the current delivered in source code, since it might be also useful for next developments



(a)

(b)

(c)

Figure 15. Messages of Non-Reciprocal, Reciprocal and all Sessions for All Cases | Session Type: 3

2.5. Conclusion

During this work the Perverted Justice dataset was corrected and enriched, while several APIs for the analysis of the dataset with more advanced statistical and machine learning techniques dataset were developed. These APIs led to some meaningful conclusions. Initially, the number of messages exchanged during a day, but also over the course of several days was extracted and the users' activity was identified. In doing so, indications of whether bots are used to exchange messages can become available. In addition, the density of the messages and well as the existence of sessions with activity only from one of the two participants can be distinguished by selecting different session durations. Finally, the time to reply to the other participants' messages can show how eager the predator and/or the victim are to continue the conversation.

Subsequently, the reciprocal and non-reciprocal sessions for all cases of the Perverted Justice dataset were investigated leading to the following conclusions:

- The percentage of non-reciprocal sessions is maintained at a relatively low level (10%-40%) as the time window between two consecutive messages is small (30sec), while it increases (10%-80%) as the time window grows. This outcome can be interpreted as follows: predators tend to insist long enough before aborting a conversation with their potential victim, even when he or she does not respond to their messages.
- However, the length of the time window seems not to affect the cumulative percentage of non-reciprocal sessions: 90% of all cases correspond to about 50% of the non-reciprocal sessions both for small or larger time windows.
- Almost all non-reciprocal sessions are characterized by a low number of exchanging messages between the victim and the predator, while reciprocal sessions present a relatively larger number of exchanging messages.

3. Identification of Datasets Including Real and Friendly Chat Dialogues

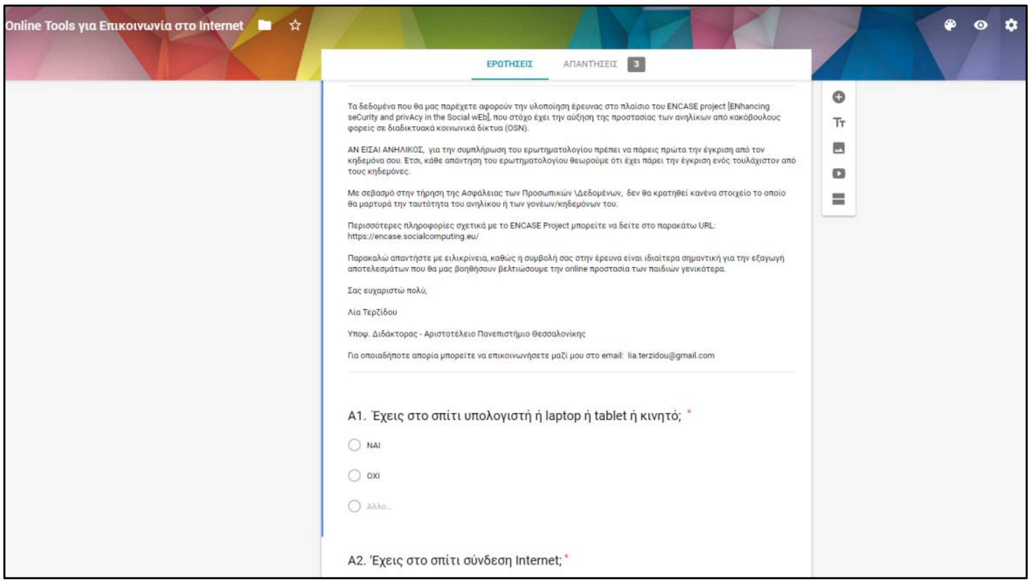
3.1. Project Description and Motivation

Subsequently, the research interest focused on the identification of available datasets including chat conversations between what can be considered benign users (i.e., users that have conversations in a friendly or collaborative context) or that include real every-day dialogues between minors (8-18 years old), in a friendly context. The datasets that were found did not meet the requirements and due to this reason, they were not utilized for further analysis.

3.2. Design of Questionnaire to Investigate the Online Communication Preferences of Minors

Given the difficulties on finding an appropriate dataset from real data of students' friendly online converse actions, the opportunity to create our own dataset from real data of students' friendly online conversations was investigated. Towards this end, as initial step, an online questionnaire was created in order to investigate which social networks do minors mostly use to chat with their friends (Figure 16). The questionnaire is available in Greek at https://goo.gl/forms/lnlaLWpYQJk21gQv2 and will be translated also in English.

The survey sample concerns students (8-17 years old) who belong to CoderDojo Thessaloniki, a volunteer community, which is part of the CoderDojo Foundation. The CoderDojo Foundation was informed about the current state regarding legal and GDPR issues on utilizing students' data stored in CoderDojo's Zen platform, which is GDPR compliant.



Τα δεδομένα που θα μας παρέχετε αφορούν την υλοποίηση έρευνας στο πλαίσιο του ENCASE project [ENhancing seCurity and privAcy in the Social wEb], που στόχο έχει την αύξηση της προστασίας των ανηλίκων από κακόβουλους φορείς σε διαδικτυακά κοινωνικά δίκτυα (OSN).

ΑΝ ΕΙΣΑΙ ΑΝΗΛΙΚΟΣ, για την συμπλήρωση του ερωτηματολογίου πρέπει να πάρεις πρώτα την έγκριση από τον κηδεμόνα σου. Έτσι, κάθε απάντηση του ερωτηματολογίου θεωρούμε ότι έχει πάρει την έγκριση ενός τουλάχιστον από τους κηδεμόνες.

Με σεβασμό στην τήρηση της Ασφάλειας των Προσωπικών \Δεδομένων, δεν θα κρατηθεί κανένα στοιχείο το οποίο θα μαρτυρά την ταυτότητα του ανηλίκου ή των γονέων/κηδεμόνων του.

Περισσότερες πληροφορίες σχετικά με το ENCASE Project μπορείτε να δείτε στο παρακάτω URL: https://encase.socialcomputing.eu/

Παρακαλώ απαντήστε με ειλικρίνεια, καθώς η συμβολή σας στην έρευνα είναι ιδιαίτερα σημαντική για την εξαγωγή αποτελεσμάτων που θα μας βοηθήσουν βελτιώσουμε την online προστασία των παιδιών γενικότερα.

Σας ευχαριστώ πολύ,

Λία Τερζίδου

Υποψ. Διδάκτορας - Αριστοτέλειο Πανεπιστήμιο Θεσσαλονίκης

Για οποιαδήποτε απορία μπορείτε να επικοινωνήσετε μαζί μου στο email: lia.terzidou@gmail.com

A1. Έχεις στο σπίτι υπολογιστή ή laptop ή tablet ή κινητό; *

○ ΝΑΙ

○ ΟΧΙ

○ Άλλο...

A2. 'Εχεις στο σπίτι σύνδεση Internet; *

Figure 16. Questionnaire to investigate the online communication preferences of minors (in Greek)

The process for the creation of a new applicable dataset includes following steps:

i. Both parents and students registered on Zen platform, will be contacted and requested to provide us the permission, based on GDPR, to use their contact data (i.e. email) from CoderDojo's Zen platform, on behalf of the ENCASE project. Towards this end, an official and detail informative letter must be prepared by the ENCASE project, to be shared with the potential subjects of the survey.

ii. The questionnaire will be shared online with the students aiming to identify the communication tools the students prefer. The analysis of the survey will identify the communication tool which will be selected for collecting the appropriate data for the dataset.

iii. Having secured all the necessary legal requirements of the project, the permission and consent of both parents and students will be requested to provide us the students' personal chat logs to create the dataset of real friendly dialogues. Towards this end, the development of a customized software is considered necessary for filtering and encrypting all personal data that may appear on students' chat logs (i.e. name-surname, phone number, post address, email, etc.).

Summarizing, the creation of a new dataset will exploit following data:

- *Existing data from the CoderDojo's Zen platform*
  - Name / Surname
  - email: To contact parents-youths
- *Private data that will be collected*

- Regarding the online communication tools and OSN usage (through the online questionnaire)
- Chat logs from kids'/youths' everyday online personal communication interactions with friends/peers, to identify patterns in dialogues conducted in a friendly manner

3.3. About CoderDojo Thessaloniki

The CoderDojo Thessaloniki club is part of the global CoderDojo movement, a global network of free, volunteer-led and community-based programming clubs for young people between 7-17 years old. CoderDojo Thessaloniki offers free and not-for-profit regular sessions for young people, where they can learn to code, build a website, create an app or a game, and explore technology in an informal, creative, and social environment. Dojos are set up, run by and taught at by volunteers. The CoderDojo Foundation was established in 2013 by CoderDojo co-founder James Whelton. CoderDojo Thessaloniki was founded by Theodouli Terzidou in 2014 and is hosted at the facilities of The University of Sheffield International Faculty, CITY College in Thessaloniki.

To attend a Dojo at CoderDojo Thessaloniki, parents and students have to register to 'Zen', the CoderDojo community platform hosted at zen.coderdojo.com. Moreover, Zen is a platform for CoderDojo community members (Champions, Mentors, Parents and Youth Attendees) to search for local Dojos, create listings for their Dojos on the publicly viewable Dojo database, issue and book tickets for their Dojos, participate in the CoderDojo web forums and manage their Dojo volunteers. The CoderDojo Foundation offers this website, including all information, tools and services available from this site to its users, conditioned upon their acceptance of all terms, conditions, policies and notices stated here. The Terms and Conditions are a legal contract between the users and the CoderDojo Foundation regarding users' use of Zen.

Zen is operated and presented to users by the CoderDojo Foundation, with registered address Dogpatch Labs, The CHQ Building, Custom House Quay, Dublin 1, Ireland. CoderDojo Foundation is an Irish Company registered under company number 524255 and a registered charity in Republic of Ireland, CHY 20812.

3.4. Translate of the Questionnaire for Online Communication Tools

The questionnaire for the investigation of minors' online communication habits and preferences was created in Greek language. It might useful to translate it also in the languages of the ENCASE partners (e.g., Italian, English, Spanish) to be shared and completed also by minors from other countries. Note that in this case the General Data Protection Regulation (GDPR) must be ensured, as well the legality of the whole research process.

3.5. Future Work

Some thoughts about future work are as follows:

i. **Research minors' online communication habits and preferences**. An initial investigation of students' online communication preferences would be useful to identify popular communications tools that children and adolescents use in their everyday life. The designed questionnaire could be utilized for this purpose.

ii. The results of this research are expected to reveal tools and issues on which is needed to focus on regarding kids’ and teens’ online safety. Moreover, one or two communication tools could be selected for further research.

iii. **Create our own suitable dataset.** Given the difficulty in finding available datasets, which include real every-day online dialogues between minors (8-18 years old), in a friendly context is proposed to create a new dataset that will meet the requirements of the ENCASE project. A suitable sample towards this end could be minors, who, and their parents, will give their consent to the storage of their personal-friendly online conversations. To ensure their anonymity, a tool could be developed to anonymously store their online dialogues (keep chat logs) for a specific period and for a specific communication tool (select one from the most popular tools, as these emerged from an initial research, as this was described above (*see i.*)).

iv. **Comparison of Perverted Justice dataset with the new created friendly Dataset.** A comparison of *Perverted Justice* dataset with another “friendly” one is needed to identify differences and patterns in their behavior, as well as to develop a decision-making procedure to identify predator from any other user.

v. **Extend the Python source code to visualize statistics for all session types**. Towards this end it has considered the existence of the *sessionType* variable, which is already available and provided by the current developed code and needs just to be adopted (iterate for all *SessionIds* from 1 to 4).

vi. **Create a Box Whisker Plot** to present visually statistics for the messages’ distribution of Non-Reciprocal sessions for each Case of the Perverted Justice dataset. Note that the needed data for the creation of the particular Box Whisker Plot have been considered and are available in the current developed code through the *case_session_stats* and *SessionsStats* data structures.

4. OSN Malicious Users Time-dependent Detection

4.1. Project Description and Motivation

This project focused on identifying, extracting and cleansing of group conversations with the purpose of extracting bidirectional friendly conversation datasets between two group participants, while a collective approach for analyzing chat conversations was followed.

Previous work conducted in the project context focused on predator-victim chat conversation and attempted to organize chat lines from the same person (predator or victim) into common behavior blocks (sessions). These blocks allowed the identification of specific patterns of behavior and how these affect the behavior or reaction of the chat responder. In this work, a thorough study of the current approaches regarding sentiment and affective analysis was performed, including their implementation using Natural Language Processing (NLP) and machine learning approaches.

The Perverted Justice[2] dataset has been utilized for the purposes of our research. This dataset does

[2] http://www.perverted-justice.com/index.php

not have non-harassment conversations that can be used to differentiate the identified patterns. Moreover, regular conversations between chat users and individuals are not publicly available and protected under privacy policies. Considering these gaps, the current work has focused on looking for other ways to obtain the desired data sets that will enhance and evaluate the predator identification methods that have been created. The proposed methods for overcoming this issue are presented in the next subsection (4.2).

In this framework, the following three approaches were investigated:

1) Synthetic data generation

Synthetic data is information that's artificially manufactured rather than generated by real-world events. Synthetic data is created algorithmically, and it is used as a stand-in for test datasets of production or operational data to train machine learning models. Synthetic data are often generated to meet specific needs or certain conditions that cannot be found in the real data. Synthetic data also fill the purpose of protecting privacy and confidentiality of real-world data, which is the main reason that we consider this approach for the acquisition of a conversation between two individuals. This can be applied to create a dataset which contains the generated results that are the possible effect scores produced by the sentiment and affective analysis.

2) Chatrooms

The main feature of chatroom conversations is that they are often moderated, so it is highly unlikely for a user to misbehave or show any signs of malicious behavior. This can ensure that the conversations that are extracted from these datasets will provide a good sample for labeling the conversations of common users. Chatroom conversations can be useful to acquire conversations between two individuals because at some level the behavior of the participants in certain situations is similar with the behavior of one-to-one conversations.

3) Datasets for chatbots training

An effective chatbot requires a massive amount of data in order to quickly solve user inquiries without human intervention. This leads to the creation of many task-oriented dialog data to train these complex systems. These Datasets can be considered as one-to-one chat conversations, as they are created with the purpose of simulating conversation between individuals. Furthermore, we can claim that these datasets don not contain conversations that can be considered offensive or malicious.

Based on the above, we focused on the first two approaches, trying to produce benign datasets that can be used to evaluate the sexual predator identification methods that were previously developed in the project. The datasets that were examined are presented in Table 2.

Table 2. Chatroom datasets identified online and obtained

	Name	Attributes	Format	Description
Chatroom Datasets	FreeCode Camp[3]	Username Text	CSV files	The files contain the posts from students, bots, moderators and contributors in the

[3] https://www.kaggle.com/freecodecamp/all-posts-public-main-chatroom

		Sent Mentions		main Gitter chatroom between 31-Dec-2014 until the first days of Dec-2017. There are around 5 million posts from near 400,000 users (estimates)
	Stack Exchange Data Dump[4]	CommentCount AnswerCount Score ViewCount Body	XML files	There are multiple files containing Posts, Comments and Post History of the StackExchange platform[5]
Chatbot training Datasets	Maluuba Frames Dataset[6]	user_id turns wizard_id labels	JSON file	A corpus of 1369 human-to-human dialogues with an average of 15 turns per dialogue.
	Cornell movie-dialogs corpus[7]	movie_lines movie_conversations	TXT files	This corpus contains a large metadata-rich collection of 220,579 conversational exchanges between 10,292 pairs of movie characters

4.2. Methodology

The work that has been conducted for the implementation of the first two proposed methods is described in this subsection.

1) Synthetic data generation

In the framework of this method, we chose to synthesize data that simulate possible affects scores that can be produced by a sentiment and affect analysis. Affects scores were selected to be generated for the reason that they are quantitative, and this makes them suitable for this approach.

Our approach for this method follows the next steps:

i. Use the “WP4 Encase Demo” API[8] to get the affect analysis for each case in json format and use the following part of the response to get the affect scores. Affect scores are provided in the following json format:

```
"ChatLogs":
{
    "Index": 0,
    "Date": "2018-11-01T12:00:00.000Z",
    "Username": "string",
    "PV": "string",
    "Text": "string",
    "Affect": {
    "anger": 0,
```

[4] https://archive.org/details/stackexchange

[5] https://stackexchange.com

[6] https://datasets.maluuba.com/Frames

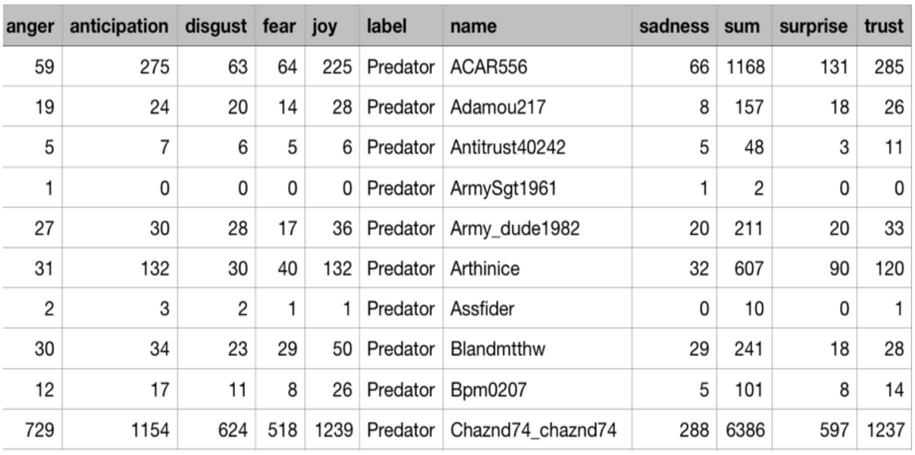
[7] http://www.cs.cornell.edu/~cristian//Cornell_Movie-Dialogs_Corpus.html

[8] WP4 ENCASE DEMO API

```
        "anticipation": 0,
        "disgust": 0,
        "sadness": 0,
        "joy": 0,
        "surprise": 0,
        "trust": 0,
        "fear": 0,
        "positive": 0,
        "negative": 0
    }
```

Sum of all affect scores for the ‘victim’ and ‘predator’ of each case and store them in two datasets, one that contains the data of the victims and one for the predators. These datasets have the following format (Table 3):

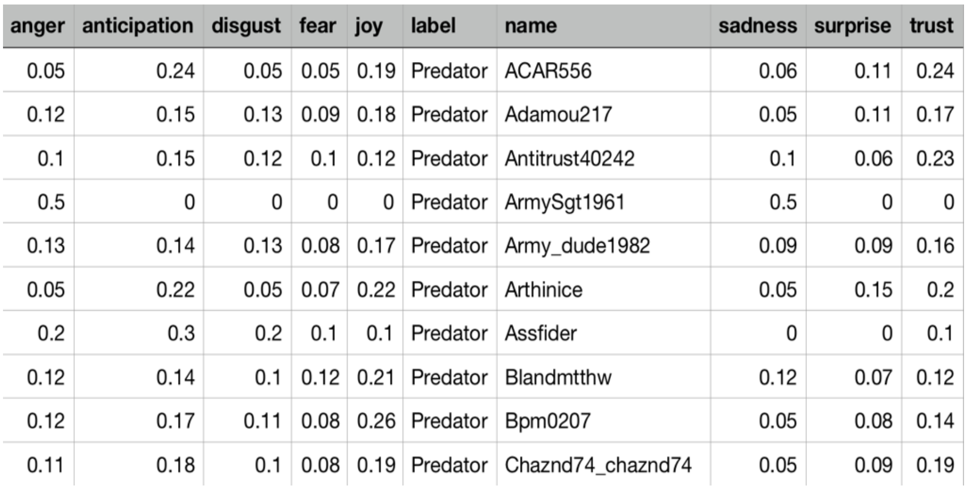
Table 3. Sum affect scores for each case



anger	anticipation	disgust	fear	joy	label	name	sadness	sum	surprise	trust
59	275	63	64	225	Predator	ACAR556	66	1168	131	285
19	24	20	14	28	Predator	Adamou217	8	157	18	26
5	7	6	5	6	Predator	Antitrust40242	5	48	3	11
1	0	0	0	0	Predator	ArmySgt1961	1	2	0	0
27	30	28	17	36	Predator	Army_dude1982	20	211	20	33
31	132	30	40	132	Predator	Arthinice	32	607	90	120
2	3	2	1	1	Predator	Assfider	0	10	0	1
30	34	23	29	50	Predator	Blandmtthw	29	241	18	28
12	17	11	8	26	Predator	Bpm0207	5	101	8	14
729	1154	624	518	1239	Predator	Chaznd74_chaznd74	288	6386	597	1237

Standardization of these 2 datasets. The standardization in our case was to convert the scores to percentages. New datasets format is depicted in Table 4.

Table 4. Percentages of affect scores for each case



anger	anticipation	disgust	fear	joy	label	name	sadness	surprise	trust
0.05	0.24	0.05	0.05	0.19	Predator	ACAR556	0.06	0.11	0.24
0.12	0.15	0.13	0.09	0.18	Predator	Adamou217	0.05	0.11	0.17
0.1	0.15	0.12	0.1	0.12	Predator	Antitrust40242	0.1	0.06	0.23
0.5	0	0	0	0	Predator	ArmySgt1961	0.5	0	0
0.13	0.14	0.13	0.08	0.17	Predator	Army_dude1982	0.09	0.09	0.16
0.05	0.22	0.05	0.07	0.22	Predator	Arthinice	0.05	0.15	0.2
0.2	0.3	0.2	0.1	0.1	Predator	Assfider	0	0	0.1
0.12	0.14	0.1	0.12	0.21	Predator	Blandmtthw	0.12	0.07	0.12
0.12	0.17	0.11	0.08	0.26	Predator	Bpm0207	0.05	0.08	0.14
0.11	0.18	0.1	0.08	0.19	Predator	Chaznd74_chaznd74	0.05	0.09	0.19

Standardization of all data in order to observe the distribution for all eight affects. All affects is considered to follow normal distribution. Indicatively the distribution of “Fear” affect, as shown in Figure 17.

- Calculation of the mean values and standard deviation for each affect
- Calculation of the mean values and standard deviation for the generated data. Using T-test procedure[9] we choose new mean values and standard deviation values that are statistically independent from the original dataset.
- Generation of the desired data using the above calculated values as input to a NumPy’s function[10] that generates random data following normal distribution

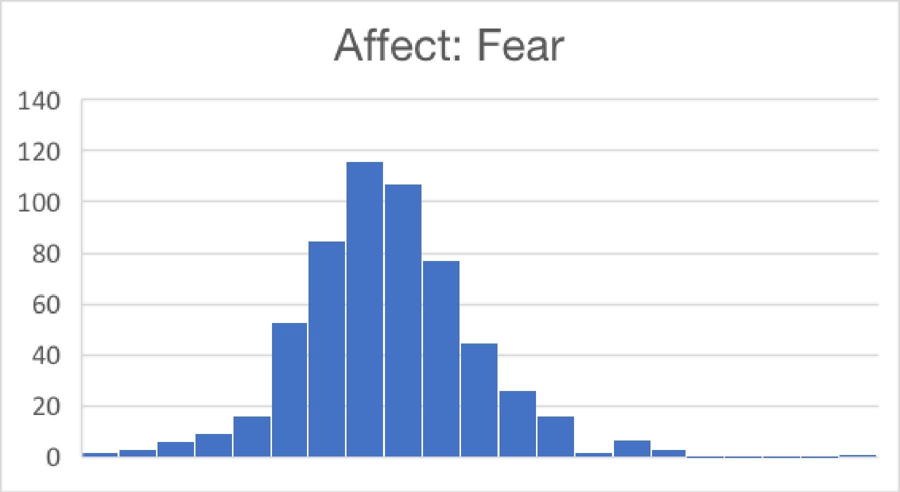


Figure 17. 'Fear' affect distribution

The next steps are: i) to compare the generated data with the original dataset to validate that they represent different groups of people, and ii) to check whether they match the analysis of

[9] http://www.quantitativeskills.com/sisa/statistics/t-test.htm

[10] https://docs.scipy.org/doc/numpy-1.15.0/reference/generated/numpy.random.normal.html

conversations that will potentially be retrieved by other methods.

2) Chatroom conversation parsing

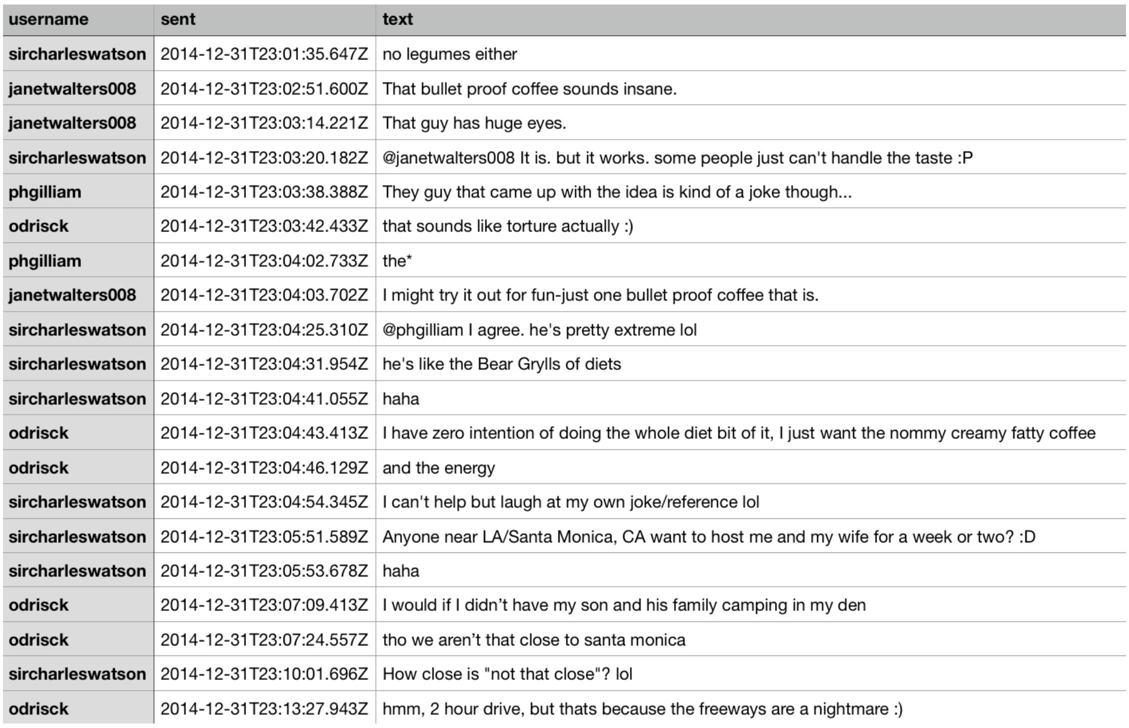
As described in the previous section, the goal is to use chatroom conversation and extract conversations between two individuals. Table 5 presents a snapshot of the chatroom that was used for examining our approach. At first glance it seems that the conversation topics are abstract. However, looking more closely, it seems that there are parts that carry out a one-to-one conversation, as is the case in Figure 18 between users 'sircharleswatson' and 'odrisck'.

In the framework of this approach we attempt to retrieve this kind of conversations between two users. Towards that goal the following steps were performed:

- Removing the records of the users that are not active. Active users are considered the users that have sent a number of messages that are over a limit. Pending further limit examination in our preliminary experiments this limit was set to 20 messages.
- Combination of consecutive messages of the same user that don’t have a time difference between them that is longer than a given time period. In our case the period was set to 1 hour.
- Determination of the criteria to be met so that a message to be considered as part of a conversation. These criteria are the following:
    a. Message contains reference to a user. eg. “Hello @User3245” or “@User3245 what’s up? ”
    b. Message is a part of a brief conversation between two users (e.g. the two user exchange a number of messages without any other user to intervene) in the chatroom. For instance:

        -**User1:** “Good morning, fine day for driving.”
        -***User2:*** “Yes, the weather looks sunny today.”
        -**User1:** “Feels hotter than yesterday.”
        -***User2:*** “That’s for sure!!”

Table 5. Chatroom snapshot

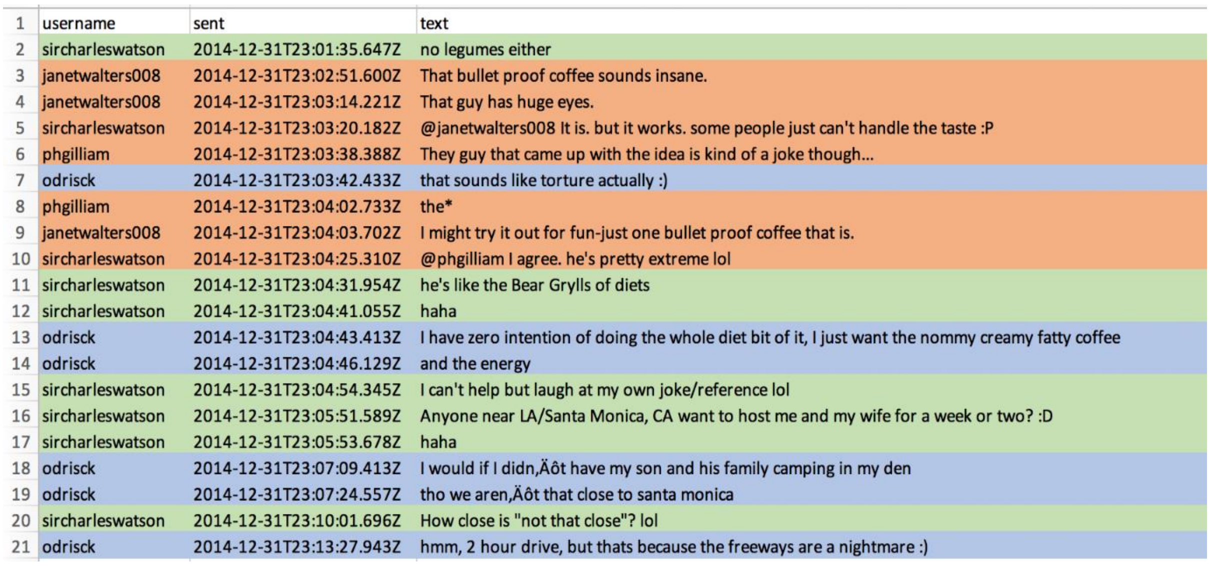


username	sent	text
sircharleswatson	2014-12-31T23:01:35.647Z	no legumes either
janetwalters008	2014-12-31T23:02:51.600Z	That bullet proof coffee sounds insane.
janetwalters008	2014-12-31T23:03:14.221Z	That guy has huge eyes.
sircharleswatson	2014-12-31T23:03:20.182Z	@janetwalters008 It is. but it works. some people just can't handle the taste :P
phgilliam	2014-12-31T23:03:38.388Z	They guy that came up with the idea is kind of a joke though...
odrisck	2014-12-31T23:03:42.433Z	that sounds like torture actually :)
phgilliam	2014-12-31T23:04:02.733Z	the*
janetwalters008	2014-12-31T23:04:03.702Z	I might try it out for fun-just one bullet proof coffee that is.
sircharleswatson	2014-12-31T23:04:25.310Z	@phgilliam I agree. he's pretty extreme lol
sircharleswatson	2014-12-31T23:04:31.954Z	he's like the Bear Grylls of diets
sircharleswatson	2014-12-31T23:04:41.055Z	haha
odrisck	2014-12-31T23:04:43.413Z	I have zero intention of doing the whole diet bit of it, I just want the nommy creamy fatty coffee
odrisck	2014-12-31T23:04:46.129Z	and the energy
sircharleswatson	2014-12-31T23:04:54.345Z	I can't help but laugh at my own joke/reference lol
sircharleswatson	2014-12-31T23:05:51.589Z	Anyone near LA/Santa Monica, CA want to host me and my wife for a week or two? :D
sircharleswatson	2014-12-31T23:05:53.678Z	haha
odrisck	2014-12-31T23:07:09.413Z	I would if I didn't have my son and his family camping in my den
odrisck	2014-12-31T23:07:24.557Z	tho we aren't that close to santa monica
sircharleswatson	2014-12-31T23:10:01.696Z	How close is "not that close"? lol
odrisck	2014-12-31T23:13:27.943Z	hmm, 2 hour drive, but thats because the freeways are a nightmare :)

4.3. Initial Results

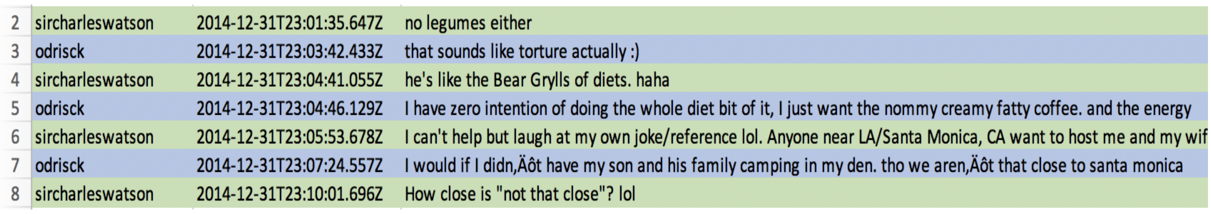
We first test our "chatroom conversation parsing" method on the FreeCode camp dataset that consists of 65500 messages, to observe some initial results and examine the potentials for this approach. Table 6 depicts how the messages in a chatroom conversation are differentiated by our approach. Messages with orange color are the ones that are from or to an inactive user and were removed from the dataset, with blue and green color are the messages of each user. Table 7 shows how the dataset changes after the process described in the previous subsection.

Table 6. Chatroom messages distinction



1	username	sent	text
2	sircharleswatson	2014-12-31T23:01:35.647Z	no legumes either
3	janetwalters008	2014-12-31T23:02:51.600Z	That bullet proof coffee sounds insane.
4	janetwalters008	2014-12-31T23:03:14.221Z	That guy has huge eyes.
5	sircharleswatson	2014-12-31T23:03:20.182Z	@janetwalters008 It is. but it works. some people just can't handle the taste :P
6	phgilliam	2014-12-31T23:03:38.388Z	They guy that came up with the idea is kind of a joke though...
7	odrisck	2014-12-31T23:03:42.433Z	that sounds like torture actually :)
8	phgilliam	2014-12-31T23:04:02.733Z	the*
9	janetwalters008	2014-12-31T23:04:03.702Z	I might try it out for fun-just one bullet proof coffee that is.
10	sircharleswatson	2014-12-31T23:04:25.310Z	@phgilliam I agree. he's pretty extreme lol
11	sircharleswatson	2014-12-31T23:04:31.954Z	he's like the Bear Grylls of diets
12	sircharleswatson	2014-12-31T23:04:41.055Z	haha
13	odrisck	2014-12-31T23:04:43.413Z	I have zero intention of doing the whole diet bit of it, I just want the nommy creamy fatty coffee
14	odrisck	2014-12-31T23:04:46.129Z	and the energy
15	sircharleswatson	2014-12-31T23:04:54.345Z	I can't help but laugh at my own joke/reference lol
16	sircharleswatson	2014-12-31T23:05:51.589Z	Anyone near LA/Santa Monica, CA want to host me and my wife for a week or two? :D
17	sircharleswatson	2014-12-31T23:05:53.678Z	haha
18	odrisck	2014-12-31T23:07:09.413Z	I would if I didn‚Äôt have my son and his family camping in my den
19	odrisck	2014-12-31T23:07:24.557Z	tho we aren‚Äôt that close to santa monica
20	sircharleswatson	2014-12-31T23:10:01.696Z	How close is "not that close"? lol
21	odrisck	2014-12-31T23:13:27.943Z	hmm, 2 hour drive, but thats because the freeways are a nightmare :)

Table 7. Chatroom snapshot after processing



2	sircharleswatson	2014-12-31T23:01:35.647Z	no legumes either
3	odrisck	2014-12-31T23:03:42.433Z	that sounds like torture actually :)
4	sircharleswatson	2014-12-31T23:04:41.055Z	he's like the Bear Grylls of diets. haha
5	odrisck	2014-12-31T23:04:46.129Z	I have zero intention of doing the whole diet bit of it, I just want the nommy creamy fatty coffee. and the energy
6	sircharleswatson	2014-12-31T23:05:53.678Z	I can't help but laugh at my own joke/reference lol. Anyone near LA/Santa Monica, CA want to host me and my wif
7	odrisck	2014-12-31T23:07:24.557Z	I would if I didn‚Äôt have my son and his family camping in my den. tho we aren‚Äôt that close to santa monica
8	sircharleswatson	2014-12-31T23:10:01.696Z	How close is "not that close"? lol

Running our method in the whole Free Code Camp dataset yields the following results:

- Out of 1215 Total Users, 1065 Users were considered as *“Inactive”* and 150 as *“Active”*
- *2477 Conversations* between two users were retrieved

The above shows that parsing a chatroom conversation can provide a large number of conversations even if the number of active users is reduced drastically compared with the total number of users.

4.4. Future Work

Future work will be directed towards the fine-tuning of the “chatroom conversation parsing” method. Our plan is to evaluate different values for the limit of inactive users and different time periods for messages to be included in a session. Then will try to determine the best parameters for the approach by comparing the different results. In addition, we plan to test our approach using other chatroom datasets, thus creating a complete dataset of two person conversations. Furthermore, we will attempt to implement all approaches so that there can be a comparison between the different methods.

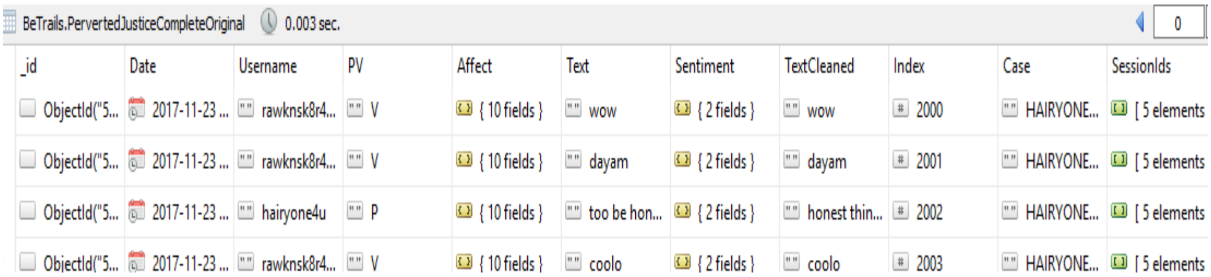
5. Towards Identifying Predator Behavior in Chat Conversations

5.1. Project Description and Motivation

The aim of this work was to discover multiple patterns indicative of predatory (sexual predators or cyberbullying) behavior over time by analyzing OSN user interactions. Advanced data mining and analytics techniques were proposed in order to leverage the OSN users' concurrent activities that indicate behavioral variations and spikes with emphasis on advancing the state of the art on anomaly detection in OSN. The purpose was the investigation of the applicability of text processing and data mining techniques, and related APIs and libraries, for the analysis of the Perverted Justice dataset towards the identification of predator behavior in chat conversation. The implementation was performed in Python language and the produced identification algorithm was integrated in the already deployed module for predator detection.

In this section, the fields of the Perverted Justice dataset are briefly described and, then, the information that is used during our experimental study is presented. Table 8 shows the structure of available information in the dataset. Each record is related to a post whose original text is kept in the field "Text". The rest fields such as the "Date", "Username", "PV", "Affect", "Sentiment", "TextCleaned", "Index", "Case", and "SessionIds" are referring to the date/time of the post, the identity (username) of the user that posts the comment, the nature of the user (P for predator, V for victim), the affect scores of the post, the sentiment scores of the post, a cleaned version of the text, the post's index, the case name and the session id as it is produced according to the time-dependent analysis, respectively. Three collections were created in MongoDB for a clear and convenient creation of the training/test sets that we used in our study. The first collection includes the original posts of each chat dialogue, i.e., the dialogues between the predators and the victims. However, due to our interest in analyzing separately the predators' from the victims' posts, we created two additional collections, one for the chats containing only the predators' posts and an additional one for the chats of the victims' posts.

Table 8. Perverted Justice dataset

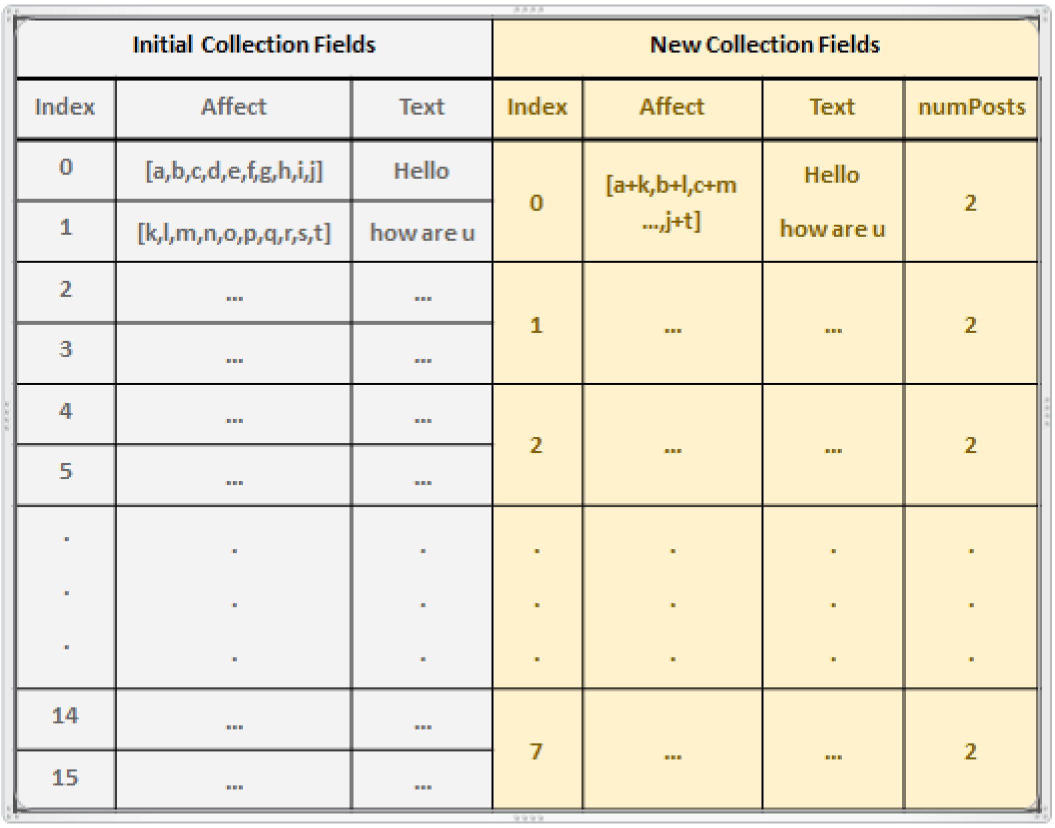


BeTrails.PervertedJusticeCompleteOriginal 0.003 sec. 0

_id	Date	Username	PV	Affect	Text	Sentiment	TextCleaned	Index	Case	SessionIds
□ ObjectId("5...	2017-11-23 ...	rawknsk8r4...	V	{ 10 fields }	wow	{ 2 fields }	wow	2000	HAIRYONE...	[5 elements
□ ObjectId("5...	2017-11-23 ...	rawknsk8r4...	V	{ 10 fields }	dayam	{ 2 fields }	dayam	2001	HAIRYONE...	[5 elements
□ ObjectId("5...	2017-11-23 ...	hairyone4u	P	{ 10 fields }	too be hon...	{ 2 fields }	honest thin...	2002	HAIRYONE...	[5 elements
□ ObjectId("5...	2017-11-23 ...	rawknsk8r4...	V	{ 10 fields }	coolo	{ 2 fields }	coolo	2003	HAIRYONE...	[5 elements

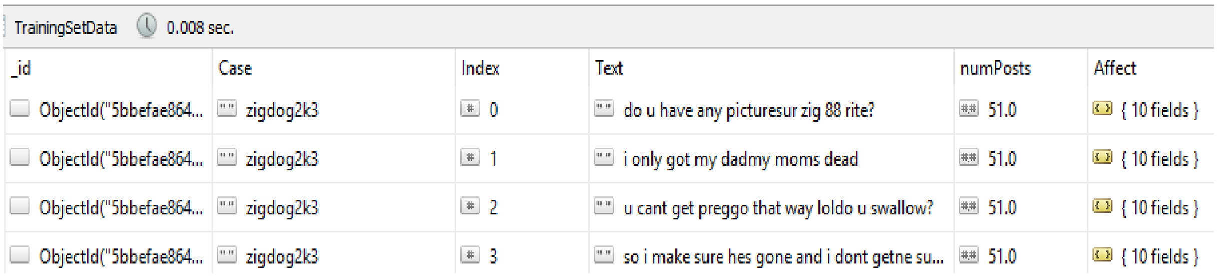
Specifically, for the creation of the first collection, we split each transcript of each case up to 8 parts based on the "Index" field of the initial dataset (see the "Index" fields in Table 9). For example, suppose a trivial case that includes 16 posts which means that the values of "Index" for each record/post of the initial dataset range in [0, 15]. We divided the chat into 8 parts by creating groups of consecutive posts. In this case, the first group of posts contains the first 2 posts (initial collection "Index"=0,1) and the "Index" field in the new collection has the value 0, the second group of posts contains the next 2 posts (initial collection "Index"=2,3) and the "Index" field in the new collection has the value 1, etc. Then, for each group of posts, we concatenated the text of the corresponding posts (see the "Text" fields in Table 9) in order to create the updated field "Text" of the new collection. Consequently, an updated "Affect" field was created which contains the sum of the initial "Affect" fields of the corresponding posts (see the "Affect" fields in Table 9)). Furthermore, we added an additional field called "numPosts" where we kept the number of posts included in the corresponding group of posts of the transcript. Finally, the field "Case" has the same use as previously. Table 10 shows the structure of the new collection.

Table 9. Example for the conversion of the initial record scheme to the new record scheme



Initial Collection Fields			New Collection Fields			
Index	Affect	Text	Index	Affect	Text	numPosts
0	[a,b,c,d,e,f,g,h,i,j]	Hello	0	[a+k,b+l,c+m …,j+t]	Hello how are u	2
1	[k,l,m,n,o,p,q,r,s,t]	how are u				
2	…	…	1	…	…	2
3	…	…				
4	…	…	2	…	…	2
5	…	…				
. . .	. . .	. . .	. . .	. . .	. . .	. . .
14	…	…	7	…	…	2
15	…	…				

Table 10. Training set information



TrainingSetData 0.008 sec.

_id	Case	Index	Text	numPosts	Affect
ObjectId("5bbefae864...	zigdog2k3	0	do u have any picturesur zig 88 rite?	51.0	{ 10 fields }
ObjectId("5bbefae864...	zigdog2k3	1	i only got my dadmy moms dead	51.0	{ 10 fields }
ObjectId("5bbefae864...	zigdog2k3	2	u cant get preggo that way loldo u swallow?	51.0	{ 10 fields }
ObjectId("5bbefae864...	zigdog2k3	3	so i make sure hes gone and i dont getne su...	51.0	{ 10 fields }

Similarly, the rest two collections, concerning the predators'/victims' posts, were created. The first one includes only the predators' posts along with their related fields as described above whereas the third one keeps the similar information only for the victims' chats. Segmenting the text files into 8 equal parts allows us to treat every segment as a separate phase, giving us the opportunity for a more detailed quantitative analysis. Of course, this type of segmentation offered us a flexible way to concatenate the segments by increasing “Index” in order to get the whole chat as one segment or to deal with fewer almost equal segments, e.g. 4 or 2.

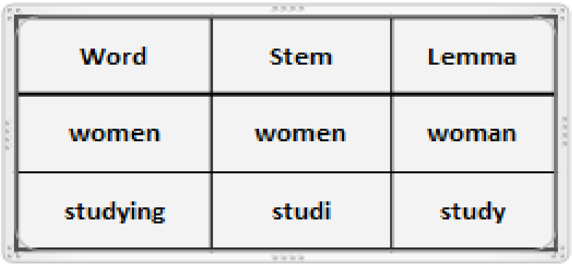
5.2. Feature Engineering

Our goal was to use both the text and the affect/sentiment scores in order to train and evaluate classification algorithms able to distinguish between predators, victims and friendly conversations. Towards that first we perform a lexicographical preprocessing in order to cleanse our dataset. Following we used several methods from the literature to create vectors of word representation for the conversations texts. These representations are then used as features for training the classification algorithms. Both the aforementioned steps are described in this section.

5.2.1. Pre-processing

Initially, each chat text was converted to lower case and english stopwords and tokens consisting only of one repeatable character (e.g., the token “mmmmmmmm”) were removed. Then, either stemming or lemmatization was applied based on the type of the text representation we use, i.e., GloVe [1] or TfIdf [2, 3, 4], respectively (both techniques of text representation will be discussed further below). Notice that we keep only the stemmed/lemmetized tokens whose length is greater than 3. The goal of both stemming and lemmatization is to reduce inflectional forms to a common base form. Stemming is the process of reducing inflected (or sometimes derived) words to their word stem by removing their derivational affixes. Lemmatization usually uses a vocabulary and morphological analysis of words to return the base or dictionary form of a word, which is known as the lemma. For a better comprehension see the following examples (Table 11).

Table 11. Stemming vs lemmatization



Word	Stem	Lemma
women	women	woman
studying	studi	study

5.2.2. Features related to the text of the (segments’) chats

In most machine learning and data mining tasks, the interest is in comparing objects in order to cluster or classify them towards distinct categories with specific characteristics that can then be used towards the classification of new incoming cases and so on. For this reason, the texts should be represented in a form that the measurement of similarity/distance among the texts can be effectively estimated. The solution to this problem is the vectorization of the text utilizing a specific text representation technique. In the following we briefly describe three such vectorization methods.

Text representation - the simple BOW technique

The simplest model, in this area, is considered to be Bag-of-Words (BOW) model. According to this model, a text (such as a sentence or a document) is represented as the bag of its words (without taking into account grammar and word order).

In the following example, we see all the stages that are required for the conversion of two simple documents into vectors based on the bag-of-words model (including a few basic preprocessing steps described earlier).

Here are the two simple text documents:

1. John likes to watch movies. Mary likes movies too.
2. John also likes to watch football games.

After the basic preprocessing stages (conversion to lower case, stop-words removal and stemming), we get:

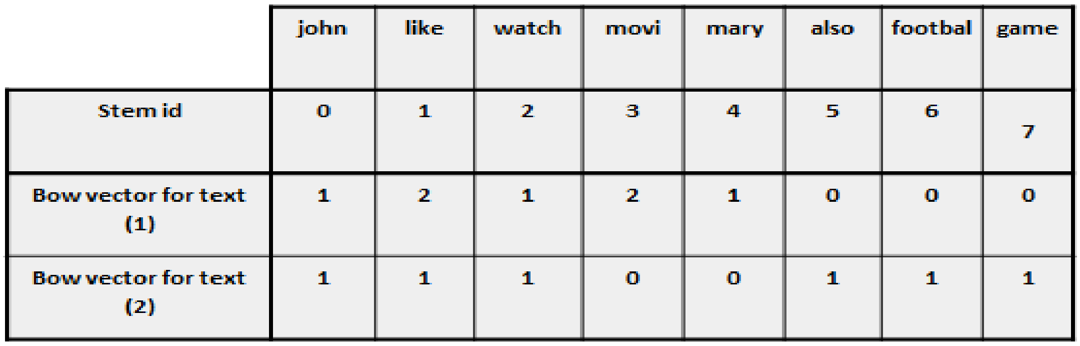
i. john like watch movi mari like movi
ii. john also like watch footbal game

Based on these two preprocessed text documents, a vocabulary set is constructed as follows:

["john","like","watch","movi","mary","also","footbal","game"]

Then, each preprocessed text is expressed as a vector which contains as values the number of occurrences of each vocabulary word (Table 12). The dimension of each vector is equal to the size of the vocabulary set, i.e., 8.

Table 12. Example of BOW text representation



	john	like	watch	movi	mary	also	footbal	game
Stem id	0	1	2	3	4	5	6	7
Bow vector for text (1)	1	2	1	2	1	0	0	0
Bow vector for text (2)	1	1	1	0	0	1	1	1

TfIdf technique

In our study, we employ a more advanced text representation model called Term Frequency-Inverse Document Frequency (TfIdf). TfIdf is a popular weight often used in information retrieval and text mining. This weight is a statistical measure used to evaluate how important a word is to a document in a collection. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the collection. Typically, the TfIdf weight is composed by two terms: the first computes the normalized Term Frequency (Tf: the number of times a word appears in a document divided by the total number of words in that document) and the second term is the Inverse Document Frequency (Idf: computed as the logarithm of the number of the documents in the corpus divided by the number of documents where the specific term appears).

Tf(t) = (Number of times term t appears in a document) / (Total number of terms in the document)
Idf(t) = log_e(Total number of documents / Number of documents with term t in it)

Let us illustrate this through a small example:
Consider a document containing 100 words wherein the words “history”, “there” and “war” appear 3, 30 and 3 times, respectively. The corresponding term frequencies for each term are the following:

Tf(history) = Tf(war) = (3 / 100) = 0.03 and Tf(there) = (30 / 100) = 0.3

Now, assume we have 10,000,000 documents and the words “history”, “there” and “war” appear in 1,000, 10,000,000, and 500 of these, respectively. Then, the inverse document frequencies for each term are the following:

Idf(history) = log(10,000,000 / 1,000) = 4,   Idf(there) = log(10,000,000 / 10,000,000) = 0 and

Idf(war) = log(10,000,000 / 500) = 4.3

Thus, the TfIdf weights are the products given below:

TfIdf(history) = 0.03 * 4 = 0.12, TfIdf(there) = 0.3 * 0 = 0 and TfIdf(war) = 0.03 * 4.3 = 0.129

which means that the most important word for the specific document is the term “war”, despite the fact that appears less times than the term “there”. Particularly, the Idf (there) is equal to 0 indicating that it is a very common (unimportant) term that occurs in all documents of the collection. Once again, the dimension of each TfIdf vector is equal to the size of the vocabulary set.

GloVe technique

TfIdf and BOW vectors are usually high-dimensional (thousands of dimensions) and sparse (most elements are zero). On the contrary, word embeddings is an alternative technique that expresses a word as a real-value dense vector of low dimension (usually about 50-300 dimensions). More specifically, word embedding methods learn a real-valued vector representation for a predefined fixed sized vocabulary from a corpus of text. The learning process is either joint with the neural network model on some task, such as document classification, or is an unsupervised process, using document statistics.

In this experimental study, the GloVe technique, which is a log-bilinear model with a weighted least-squares objective, was used. The main intuition underlying the model is the simple observation that ratios of word-word co-occurrence probabilities have the potential for encoding some form of meaning. The training objective of GloVe is to learn word vectors such that their dot product equals the logarithm of the words' probability of co-occurrence.

To sum up, GloVe word embeddings are used for various NLP applications such as part-of-speech tagging, information retrieval, question answering etc. However, it is quite a troublesome work to prepare word embeddings: we had to download large-scale data, preprocessed it, learnt it over a long time, checked the result and perform many hyperparameters tuning. So, as a first step we used the 300d pre trained word vectors that were created by training on Wikipedia 2014 + Gigaword 5 which have 400000 vocabulary size, uncased and are available here by Stanford.

The GloVe representation for a chat segment was produced by averaging (element-wise) the pre trained word vectors of the words that appeared in the corresponding text segment. We should not omit to say that in case of the TfIdf text representation we applied stemming, whereas in case of the GloVe text representation we preferred to use lemmatization, as it is more likely to meet the word lemmas than the stems in the vocabulary of the 400000 words.

5.2.3. Features related to the affects and number of posts of the (segments’) chats

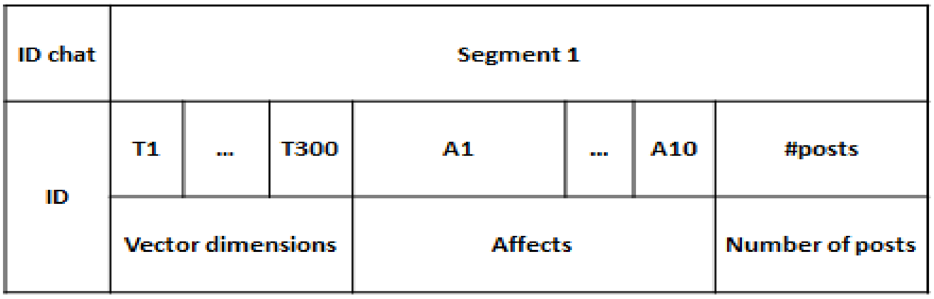
Additionally, we used the affects as features (i.e., 10 additional features). Particularly, we normalized the affects’ scores dividing each one by the number of posts that correspond to the specific chat segment. Finally, the number of posts of the chat segment is the last feature that was used.

5.3. Dataset Description

In this section, we give the overall picture of the training/test set. In the simple case that the chats are not separated into parts, each one chat consists of one segment, i.e., the Segment 1 (see Table 13). For example, when we adopted the GloVe text representation technique, each instance of the training/test set has 311 features: the first 300 correspond to the vector’s dimensions (T1-T300 features), the next 10 features to the (normalized by the number of chat’s posts) affects scores (A1-A10 features), and the last one to the number of chat’s posts (#posts), to cover for temporal behavioral characteristics.

We also provide an example where each chat is divided into two parts, i.e., the chat consists of two segments, i.e., the Segment 1 and Segment 2 (see Table 14). For every segment the features previously discussed are computed, separately. In other words, the concatenation of the two feature vectors produces the final feature vector of each training/test set instance (chat text).

Table 13. Training/test set features - Version 1



ID chat	Segment 1						
ID	T1	...	T300	A1	...	A10	#posts
	Vector dimensions			Affects			Number of posts

Table 14. Training/test set features - Version 2

ID chat	Segment 1							Segment 2						
ID	T1	...	T300	A1	...	A10	#posts	T’1	...	T’300	A’1	...	A’10	#posts’
	Vector dimensions			Affects			Number of posts	Vector dimensions			Affects			Number of posts

In cases where we split each chat up to four or eight parts, i.e., each chat consists of four or eight segments, respectively the corresponding feature vectors of the training/test set instances were generated in a similar way.

5.4. Experimental Study

In this section we briefly describe the experimental work performed in the context of this study.

5.4.1. One-class classification

One-Class SVM is particularly useful in scenarios where you have a lot of "normal" data and not many cases of the anomalies you are trying to detect. For example, if you need to detect fraudulent transactions, you might not have many examples of fraud that you could use to train a typical classification model, but you might have many examples of good transactions. SVMs are supervised learning models that analyze data and recognize patterns, and that can be used for both classification and regression tasks. Typically, the SVM algorithm is given a set of training examples labeled as belonging to one of two classes. An SVM model is based on dividing the training sample points into separate categories by as wide a gap as possible, while penalizing training samples that fall on the wrong side of the gap. The SVM model then makes predictions by assigning points to one side of the gap or the other.

Sometimes oversampling is used to replicate the existing samples so that you can create a two-class model, but it is impossible to predict all the new patterns of fraud or system faults from limited examples. Moreover, collection of even limited examples can be expensive. Therefore, in one-class SVM, the support vector model is trained on data that has only one class, which is the “normal” class. It infers the properties of normal cases and from these properties can predict which examples are unlike the normal examples. This is useful for anomaly detection because the scarcity of training examples is what defines anomalies: that is, typically there are very few examples of the network intrusion, fraud, or other anomalous behavior.

Experimental setup and results

We trained/tuned/evaluated the One-Class SVM (OC-SVM) model on the predators’ chat texts (training set) using 10-fold CV. We also tuned/evaluated the model using the victims chat texts (test set). We experimented with various feature sets as they are described in detail in Subsection 6.1.

In Table 15, we give the most interesting (highest) precision and recall scores on the predators (PPrescision-PRecall, 10-fold CV on the predators’ training set) and the victims (VPrescision-VRecall, victims as test set) achieved during the experimental study. We see that the performance is identical either we use GloVe features or not. Probably, this happens due to the high frequency of some words both in predators’ and victims’ chats. We also notice that the model built on the training set version that uses the whole chats (not divided chats in segments) performs better, whereas the more the segments of the chats in the training/test set, the lower the recall score on predators.

Table 15. Experimental study results

Text Representation	#segments per chat	Feature set	OC-SVM params (kernel-nu-gamma)	PPrecision	PRecall	VPrecision	VRecall
GloVe	1	vector+affects+#posts	sigmoid-0.5-0.001	1.00	0.75	1.00	1.00
GloVe	2	vector+affects+#posts	sigmoid-0.5-0.01	0.70	0.69	1.00	1.00
GloVe	4	vector+affects+#posts	sigmoid-0.5-0.01	0.70	0.68	1.00	1.00
GloVe	8	vector+affects+#posts	sigmoid-0.5-0.01	0.70	0.60	1.00	1.00
-	1	affects+#posts	sigmoid-0.5-0.001	1.00	0.75	1.00	1.00
-	2	affects+#posts	sigmoid-0.5-0.01	0.70	0.69	1.00	1.00
-	4	affects+#posts	sigmoid-0.5-0.01	0.70	0.68	1.00	1.00
-	8	affects+#posts	sigmoid-0.5-0.01	0.70	0.60	1.00	1.00
TfIdf	1	vector+affects+#posts	poly-0.5-0.001	1.00	0.50	1.00	0.51

5.4.2. Community Detection

The Louvain community detection algorithm [5] was applied on the predators’ and victims' chats. The nodes of the undirected graph are the predators and the victims, while the edge weights of the graph are calculated based on the following score:

$$1 - (EuclideanDist(vector_i, vector_j)/max)$$

where EuclideanDist($vector_i$, $vector_j$) is the Euclidean distance between the corresponding vectors of each pair of nodes (i, j) with i≠j, and max is the maximum Euclidean distance among the node pairs. A quite satisfying number of well-separated communities were detected. However, the communities include both predators and victims in almost equal percentages.

5.5. Software

We utilized the Gensim python library to represent the chat texts as TfIdf vectors and NLTK, which offers a variety of text processing libraries for stemming, lemmatization etc. We used the OneClassSVM class from sklearn python library to perform one-class classification. For the community detection experimental study, we used Gephi which is an open-source network analysis and visualization software package.

5.6. Conclusion

In this work, we tried to distinguish the predators’ from the victims’ chats utilizing data mining and text analytics techniques. Our empirical study offers evidence that the one-class classification approach can infer the properties of the predators’ chats and from these properties can predict which examples are unlike the predators’ chats, i.e., victims chats.

5.7. Future Work

In the near future we intend to build on top of this work, and perform a better preprocessing of the chat texts by escaping HTML characters, decoding data, handling the apostrophe/slang occurrences, extending the stop-words’ list, splitting attached words and standardizing words. We would also like to apply feature selection approaches and re-run the one-class classification experiments on the improved training/test sets.

5.8. Section References

[1] Jeffrey Pennington, Richard Socher, Christopher D. Manning. Glove: Global Vectors for Word Representation. EMNLP 2014: 1532-1543

[2] Spärck Jones K. (1972). A Statistical Interpretation of Term Specificity and Its Application in Retrieval. Journal of Documentation. 28: 11–21.

[3] Luhn Hans Peter (1957). A Statistical Approach to Mechanized Encoding and Searching of Literary Information. IBM Journal of Research and Development. 1 (4): 309–317.

[4] Manning C.D., Raghavan P., Schutze H. (2008). Scoring, term weighting, and the vector space model. Introduction to Information Retrieval. p. 100. doi:10.1017/CBO9780511809071.007. ISBN 978-0-511-80907-1.

[5] Blondel Vincent D., Guillaume Jean-Loup, Lambiotte Renaud, Lefebvre Etienne. (2008). Fast unfolding of communities in large networks. Journal of Statistical Mechanics: Theory and Experiment. 2008 (10): P10008.

6. Early Cyberbullying detection using emotion recognition – ongoing work

6.1. Project Description and Motivation

In this task, we aim to detect cyberbullying behavior against minors. Our approach is through the emotional state of the minor. If a minor is angry/sad/frustrated this can be an early indication of being cyber bullied. In this task, we formulate, train and deploy a machine learning algorithm to predict three emotional states based on the minor's OSN conversation. The emotions in question are anger, sadness and frustration and their predictions will give an early indication of cyberbullying towards the minor. To achieve it, we developed an innovative machine learning model that efficiently captures the correlation of the conversation advancement with the emotional state of the minor.

6.2. Methodologies and tools

The machine learning model was conceptualized with the purpose to reduce the ambiguity of the utterances (spoken sentences in text format) underlying emotions via including the past utterances as additional information. The model is formulated and trained in Tensorflow and its functionality is utilized in the ENCASE framework using the Falcon software.

Due to the nature of the task, we have given special attention to make the model fast, accurate and real-time applicable. To this end, the model can predict the minor's emotions live and on the conversation progress so far. There is no limitation on what the length of the conversation can be which is mainly due to the usage of RNNs and their inherent nature to support variable sequence length.

In more detail, the model formulation consists of an Bidirectional RNN for encoding the spoken sentence (list of words) of each speaker into a data representation whose output entails the dependencies between the words in the sentence. This representation is followed by, an RNN for encoding the conversation (list of sentences) into a data representation that entails the dependencies between the sentences in the conversation. The innovation lies in the usage of a self-attention mechanism to infer the importance of each sentence to the emotion in question. This information is used to derive more accurately results regarding the emotional state of the minor.

In conjunction, the model includes an embedding layer, before the Bidirectional RNN, for the representation of the words to numerical vectors. The whole process is trained into an end to end manner that renders the model flexible enough to capture the dependencies between the conversation and emotions. Specifically we trained the model on machine learning servers provided by TID and we used the IEMOCAP dataset for training. The dataset consists of annotated conversations with each sentence manually annotated by three annotators with its corresponding emotion.

6.3. Conclusion

Our innovative machine learning model has managed to improve the state of the art in the predictions of emotions in the IEMOCAP dataset. This has shown the importance of the usage of past utterances to distinguish the emotional state of the speaker.

6.4. Future Work

The emotional states are an indirect way of detecting Cyberbullying and in our demos we are using heuristic rules to define the possibility of the minor being cyberbullied. For instance, if he/she is sad and frustrated above a specific threshold. The applicability of the model in the OSN domain is under investigation and the logs derived from the usage of the model will provide valuable insights for its further improvement. Furthermore, we are considering the possibility of forming a dataset to assist further research on the emotional results of cyberbullying.

7. Quantitative Approach to Understanding Online Antisemitism

7.1. Project Description and Motivation

A new wave of growing antisemitism, driven by fringe Web communities, is an increasingly worrying presence in the socio-political realm. The ubiquitous and global nature of the Web has provided tools used by these groups to spread their ideology to the rest of the Internet. Although the study of antisemitism and hate is not new, the scale and rate of change of online data has impacted the efficacy of traditional approaches to measure and understand this worrying trend.

In this work, we present a large-scale, quantitative study of online antisemitism. We collect hundreds of million comments and images from alt-right Web communities like 4chan's Politically Incorrect board (/pol/) and the Twitter clone, Gab. Using scientifically grounded methods, we quantify the escalation and spread of antisemitic memes and rhetoric across the Web. We find the frequency of antisemitic content greatly increases (in some cases more than doubling) after major political events such as the 2016 US Presidential Election and the "Unite the Right" rally in Charlottesville. Furthermore, this antisemitism appears in tandem with sharp increases in white ethnic nationalist content on the same communities. We extract semantic embeddings from our corpus of posts and demonstrate how automated techniques can discover and categorize the use of antisemitic terminology. We additionally examine the prevalence and spread of the antisemitic "Happy Merchant" meme, and in particular how these fringe communities influence its propagation to more mainstream services like Twitter and Reddit.

Taken together, our results provide a data-driven, quantitative framework for understanding online antisemitism. Our open and scientifically grounded methods serve as a framework to augment current qualitative efforts by anti-hate groups, providing new insights into the growth and spread of antisemitism online.

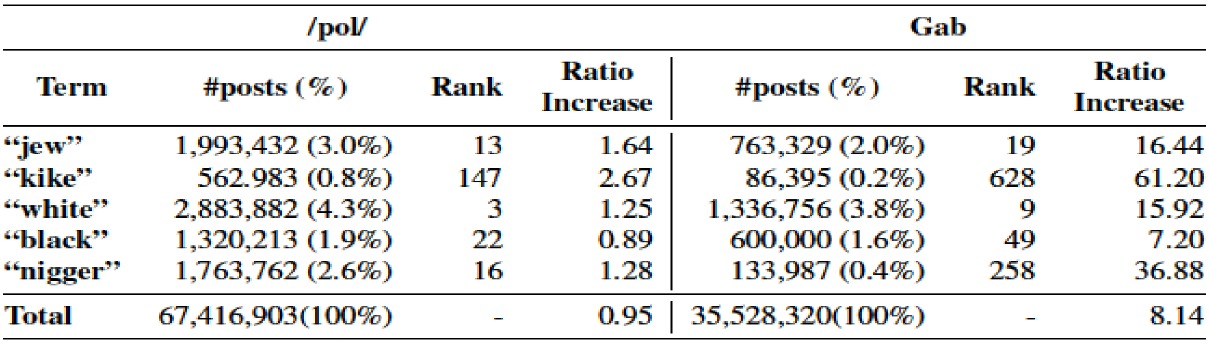
7.2. Results

In this section, we present our temporal analysis that shows the use of racial slurs over time on Gab and /pol/, our text-based analysis that leverages word2vec embeddings [22] to understand the use of text with respect to ethnic slurs, and our memetic analysis that focuses on the propagation of the anti-Semitic Happy Merchant meme. Finally, we present our influence estimation findings that shed light on the influence that Web communities have on each other when considering the dissemination of antisemitic memes.

7.2.1. Temporal Analysis

Anecdotal evidence reports escalating racial and ethnic hate propaganda on fringe Web communities [25]. To examine this, we study the prevalence of some terms related to ethnic slurs on /pol/ and Gab, and how they evolve over time. We focus on five specific terms: “jew,” “kike,” “white,” “black,” and “nigger.” We limit our scope to these because while they are notorious for ethnic hate for many groups, these specific words ranked among the most frequently used ethnic terms on both communities. Table 16 reports the overall number of posts that contain these terms in both Web communities, their rank in terms of raw number of appearances in our dataset, as well as the increase in the use of these terms between the beginning and end of our datasets. Also, Figure 18 and Figure 19 plots the use of these terms over time, binned by day, and averaged over a rolling window to smooth out small-scale fluctuations. We observe that terms like “white” and “jew” are extremely popular in both Web communities; 3rd and 13th respectively in /pol/, while in Gab they rank as the 9th and 19th most popular words, respectively. We see a similar level of popularity for ethnic racial slurs like “nigger” and “kike,” especially on /pol/; they are the 16th and 147th most popular words in terms of raw counts. Note that /pol/ has a vocabulary 1.5x times larger than that of Gab (see Text Analysis below). These findings highlight that both /pol/ and Gab users habitually and increasingly engage in discussions about ethnicity and use targeted hate speech.

Table 16. Number of posts, and their respective percentage in the dataset, for the terms “jew,” “kike,” “white,” “black,” and “nigger”



	/pol/			Gab		
Term	#posts (%)	Rank	Ratio Increase	#posts (%)	Rank	Ratio Increase
“jew”	1,993,432 (3.0%)	13	1.64	763,329 (2.0%)	19	16.44
“kike”	562.983 (0.8%)	147	2.67	86,395 (0.2%)	628	61.20
“white”	2,883,882 (4.3%)	3	1.25	1,336,756 (3.8%)	9	15.92
“black”	1,320,213 (1.9%)	22	0.89	600,000 (1.6%)	49	7.20
“nigger”	1,763,762 (2.6%)	16	1.28	133,987 (0.4%)	258	36.88
Total	67,416,903(100%)	-	0.95	35,528,320(100%)	-	8.14

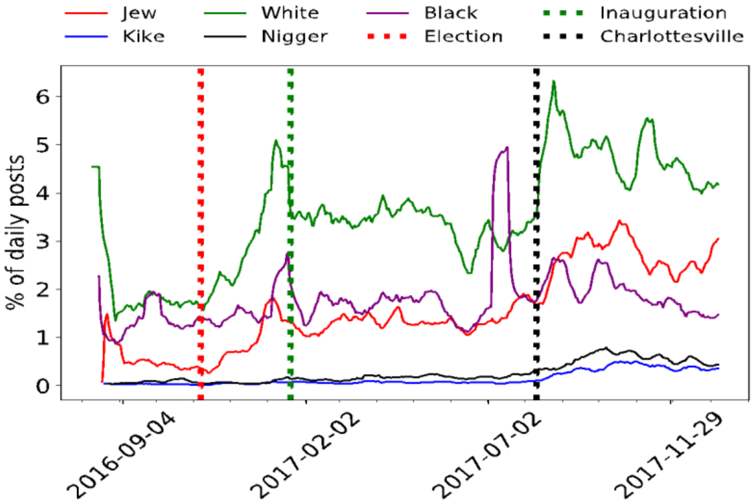


Figure 18. Use of ethnic racial terms and slurs over time on /pol/

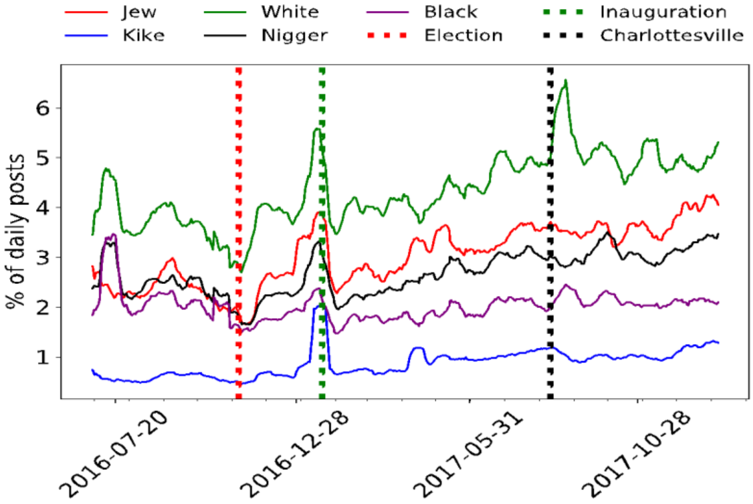


Figure 19. Use of ethnic racial terms and slurs over time on Gab

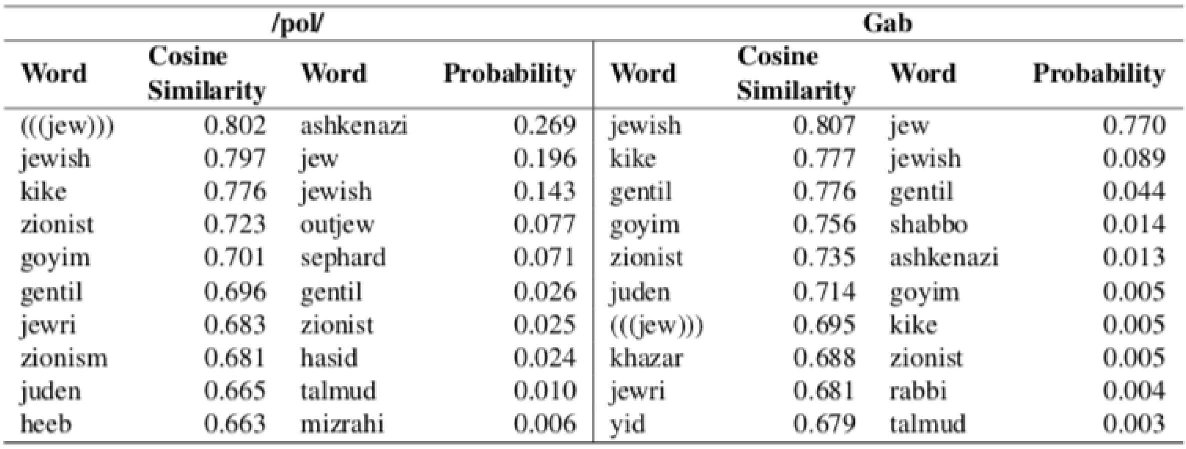
We also find an increasing trend in the use of most ethnic terms; the number of posts containing each of the terms except “black” increases, even when normalized for the increasing number of posts on the network overall. Interestingly, among the terms we examine, we observe that the term “kike” shows the greatest increase in use for both /pol/ and Gab, followed by “jew” on /pol/ and “nigger” on Gab. Also, it is worth noting that ethnic terms on Gab have a greater increase in the rate of use when compared to /pol/ (cf. ratio of increase for /pol/ and Gab in Table 16). Furthermore, by looking at Figure 19 we find that by the end of our datasets, the term “jew” appears in 4.0% of /pol/ daily posts and 3.1% of the Gab posts, while the term “nigger” appears in 3.4% and 0.6% of the daily posts on /pol/ and Gab, respectively. The latter is particularly worrisome for anti-black hate, as by the end of our datasets the term “nigger” on /pol/ overtakes the term “black” (3.4% vs 1.9% of all the daily posts). Taken together, these findings highlight that most of these terms are increasingly popular within these fringe Web communities, hence emphasizing the need to study the use of ethnic identity terms over time. We note major fluctuations in the use of ethnic terms over time, and one reasonable assumption is that these fluctuations happen due to real-world events.

7.2.2. Text analysis

We hypothesize that ethnic terms (e.g., "jew" and "white") are strongly linked to antisemitic and white supremacist sentiments. To test this, we use word2vec, a two-layer neural network that generate word representations as embedded vectors [22]. Specifically, a word2vec model takes as an input a large corpus of text and generates a multi-dimensional vector space where each word is mapped to a vector in the space (also called an embedding). The vectors are generated in such way that words that share similar contexts tend to have nearly parallel vectors in the multi-dimensional vector space. Given a context (list of words appearing in a single block of text), a trained word2vec model also gives the probability that each other word will appear in that context. By analyzing both these probabilities and the word vectors themselves, we are able to map the usage of various terms in our corpus.

We train two word2vec models; one for the /pol/ dataset and one for the Gab dataset. First, as a pre-processing step, we remove stop words (such as "and," "like," etc.) and punctuation from each post. We also perform stemming for the words in each post. Then, using the words of each post we train our word2vec models with a context window equal to 7 (defines the maximum distance between the current and the predicted words during the generation of the word vectors). Also, we consider only words that appear at least 500 times in each corpus, hence creating a vocabulary of 31,337 and 20,115 stemmed words for /pol/ and Gab, respectively. Next, we use the generated word embeddings to gain a deeper understanding of the context in which certain terms are used. We measure the "closeness" of two terms (i and j) by generating their vectors from the word2vec models (hi and hj) and calculating their cosine similarity (cos θ(h1, h2)). Furthermore, we use the trained word2vec models to predict a set of candidate words that are likely to appear in the context of a given term.

Table 17. Top ten similar words to the term "jew" and their respective cosine similarity



/pol/				Gab			
Word	Cosine Similarity	Word	Probability	Word	Cosine Similarity	Word	Probability
(((jew)))	0.802	ashkenazi	0.269	jewish	0.807	jew	0.770
jewish	0.797	jew	0.196	kike	0.777	jewish	0.089
kike	0.776	jewish	0.143	gentil	0.776	gentil	0.044
zionist	0.723	outjew	0.077	goyim	0.756	shabbo	0.014
goyim	0.701	sephard	0.071	zionist	0.735	ashkenazi	0.013
gentil	0.696	gentil	0.026	juden	0.714	goyim	0.005
jewri	0.683	zionist	0.025	(((jew)))	0.695	kike	0.005
zionism	0.681	hasid	0.024	khazar	0.688	zionist	0.005
juden	0.665	talmud	0.010	jewri	0.681	rabbi	0.004
heeb	0.663	mizrahi	0.006	yid	0.679	talmud	0.003

We first look at the term "jew." Table 17 reports the top ten most similar words to the term "jew" along with their cosine similarity, as well as the top ten candidate words and their respective probability. By looking to the most similar words, we observe that on /pol/ "(((jew)))" is the most similar term (cosθ = 0.80), while on Gab is the 7th most similar term (cos θ = 0.69). The triple parentheses is a widely used, antisemitic construction that calls attention to supposed secret Jewish involvement and conspiracy [24]. Slurs like "kike," which is historically associated with general ethnic disgust, rank similarly (cos θ = 0.77 on both /pol/ and Gab). This suggests that on both Web communities, the term "jew" itself is closely related to classical antisemitic contexts. When digging deeper, we note that "goyim" is the 5th and 4th most similar term to "jew," in /pol/ and Gab, respectively. "Goyim" is the plural of "goy," and while its original meaning is just "non-jews," modern usage tends to have a derogatory nature [27]. On fringe Web communities it is used to emphasize the "struggle" against Jewish conspiracy by preemptively assigning Jewish hostility to non-Jews as in "The Goyim Know" meme [19]. It is also commonly used in a dismissive manner toward community members; a typical attacker will accuse a user he disagrees with of being a "good goy," [15] a meme implying obedience to a supposed Jewish elite conspiracy. When looking at the set of candidate words, given the term "jew," we find the candidate word "ashkenazi" (most likely on /pol/ and 5th most likely on Gab), which refers to a specific subset of the Jewish community. Interestingly, we note that the term "jew" exists in the set of most likely words (among the top two for both communities) indicating that /pol/ and Gab users abuse the term "jew" by posting messages that include the term "jew" multiple times in the same sentence. We also note that this has a higher probability of happening on Gab rather than /pol/ (cf. probabilities for candidate word "jew" in Table 19).

To better show the connections between words similar to "jew," Figure 20 demonstrates the words associated with "jew" on /pol/ as a graph, where nodes are words obtained from the word2vec model, and the edges are weighted by the cosine distances between the words (obtained from the trained word2vec models). We extract the graph by finding the most similar words (cutoff at 0.4 cosine distance value), and then we take the 2-hop ego network around "jew. In this graph the size of a node is proportional to its degree (i.e., the number of other nodes it is directly connected to); the color of a node is based on the community it is a member of; and the entire graph is visualized using a layout algorithm that takes edge weights into account (i.e., nodes with similar words will be closer in the visualization). Note that the cosine distance is the additive inverse of the cosine similarity between two words, and we use it to demonstrate the distance between nodes in our graph. The graph visualizes the two-hop ego network [1] from the word "jew," which includes all the nodes that are either directly connected or connected through an intermediate node to the "jew" node. We consider two nodes to be connected if their corresponding word vectors have a cosine distance that is less or equal to a predefined threshold. To select only the most important connections we should select a very small percentage, therefore, we elect to set this threshold to 0.4, which corresponds to keeping only 0.2% of all possible connections (cosine distances). To identify the structure and communities in our graph, we run the community detection heuristic presented in [4], and we paint each community with a different color. Finally, the graph is layed out with the ForceAtlas2 algorithm [10], which takes into account the weight of the edges when laying out the nodes in the 2-dimensional space.

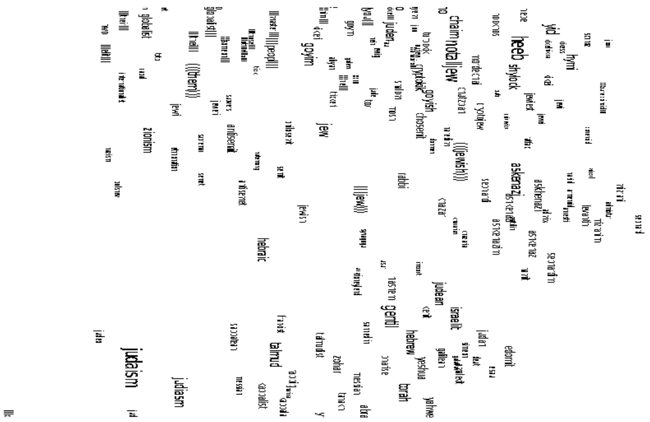


Figure 20. Graph representation of the words associated with “jew” on /pol/

This visualization reveals the existence of historically salient antisemitic terms, as well as newly invented slurs, as the most prominent associations to the word “jew.” We also note communities forming distinct themes. Keeping in mind that proximity in the visualization implies contextual similarity, we note two close, but distinct communities of words which portray Jews as a morally corrupt ethnicity on the one hand (green nodes), and as powerful geopolitical conspirators on the other (blue). Notably the blue community connects canards of Jewish political power to anti-Israel and anti-Zionist slurs. The three, more distant communities document /pol/’s interest in three topics: The obscure details of ethnic Jewish identity (grey), Kabbalistic and cryptic Jewish lore (orange), and religious, or theological topics (pink).

Table 18. Top ten similar words to the term “white” and their respective cosine similarity

/pol/				Gab			
Word	Cosine Similarity	Word	Probability	Word	Cosine Similarity	Word	Probability
huwhit	0.789	supremacist	0.494	black	0.713	supremacist	0.827
black	0.771	supremaci	0.452	huwhit	0.703	supremaci	0.147
(((white)))	0.754	supremist	0.008	nonwhit	0.684	genocid	0.009
nonwhit	0.747	male	0.003	poc	0.669	helmet	0.004
huwit	0.655	race	0.002	caucasian	0.641	nationalist	0.003
hwite	0.655	supremecist	0.002	whitepeopl	0.625	hous	0.003
whiteeuropean	0.644	nationalist	0.002	dispossess	0.624	privileg	< 0.001
hispan	0.631	genocid	0.002	indigen	0.602	male	< 0.001
asian	0.628	non	0.001	negroid	0.599	knight	< 0.001
brownblack	0.627	guilt	0.001	racial	0.595	non	< 0.001

We next examine the use of the term “white.” We hypothesize that this term is closely tied to ethnic nationalism. To provide insight for how “white” is used on /pol/ and Gab, we use the same analysis as described above for the term “jew.” Table 18 shows the top ten similar words to “white” and the top ten most likely words to appear in the context of “white.” When looking at the most similar terms, we note the existence of “huwhite” (cos θ = 0.78 on /pol/ and cos θ = 0.70 on Gab), a pronunciation of “white” popularized by the YouTube videos of white supremacist, Jared Taylor [26]. “Huwhite” is a particularly interesting example of how the alt-right adopts certain language, even language that is seemingly derogatory towards themselves, in an effort to further their ideological goals. We also note the existence of other terms referring to ethnicity, such the terms "black" (cos θ = 0.77 on /pol/ and cos θ = 0.71 on Gab), "whiteeuropean" (cos θ = 0.64 on /pol/), and "caucasian" (cos θ = 0.64 on Gab). Interestingly, we again note the presence of the triple parenthesis "(((white)))" term on /pol/ (cos θ = 0.75), which refers to Jews who conspire to disguise themselves as white. When looking at the most likely candidate words, we find that on /pol/ the term "white" is linked with "supremacist," "supremacy," and other ethnic nationalism terms.

The same applies on Gab with greater intensity as the word "supremacist" has a substantially larger probability of occurring compared to the probability obtained by the /pol/ model. To provide more insight into the contexts and use of "white" on /pol/ we show its most similar terms and their nearest associations in Figure 21 (using the same approach as for "jew"). We find seven different communities that evidence identity politics alongside themes of racial purity, miscegenation, and political correctness. These communities correspond to distinct ethnic and gender themes, like Hispanics (green), Blacks (orange), Asians (teal), and women (pink). The central community (grey) displays terms relating to whiteness with notable themes of ethnic nationalism. The final two communities relate to concerns about race-mixing (turquoise) and a prominent pink cluster that intriguingly, references terms related to left-wing political correctness [5], such as microaggression and privilege (violet).

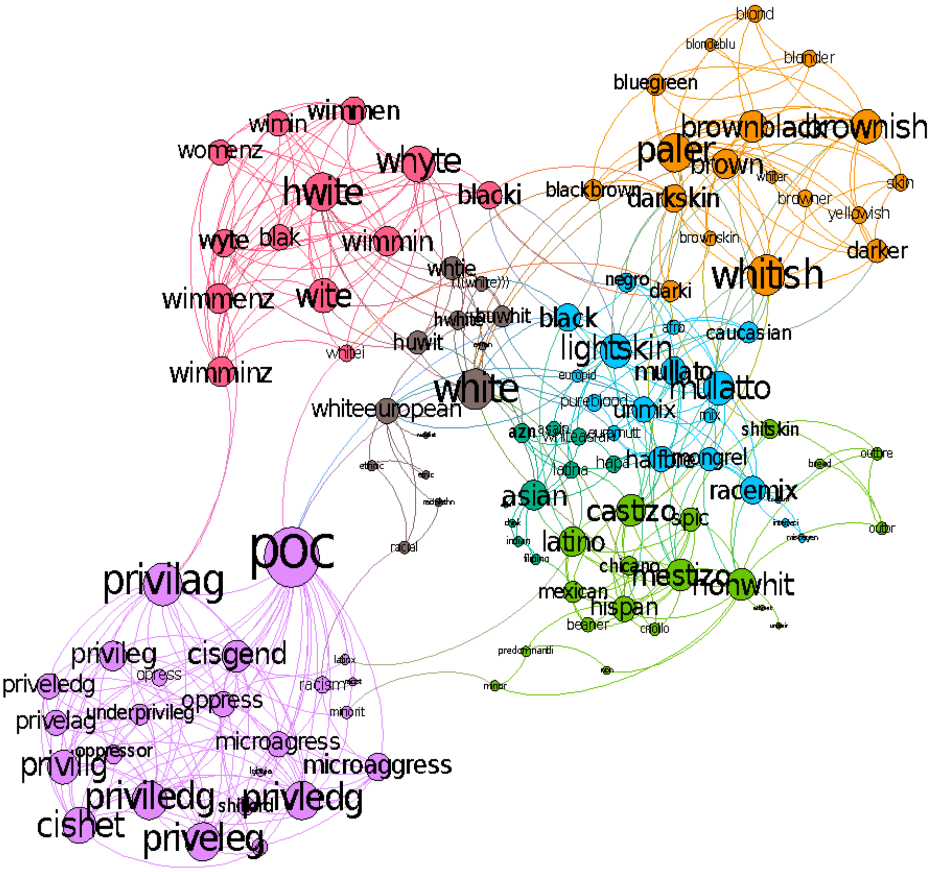
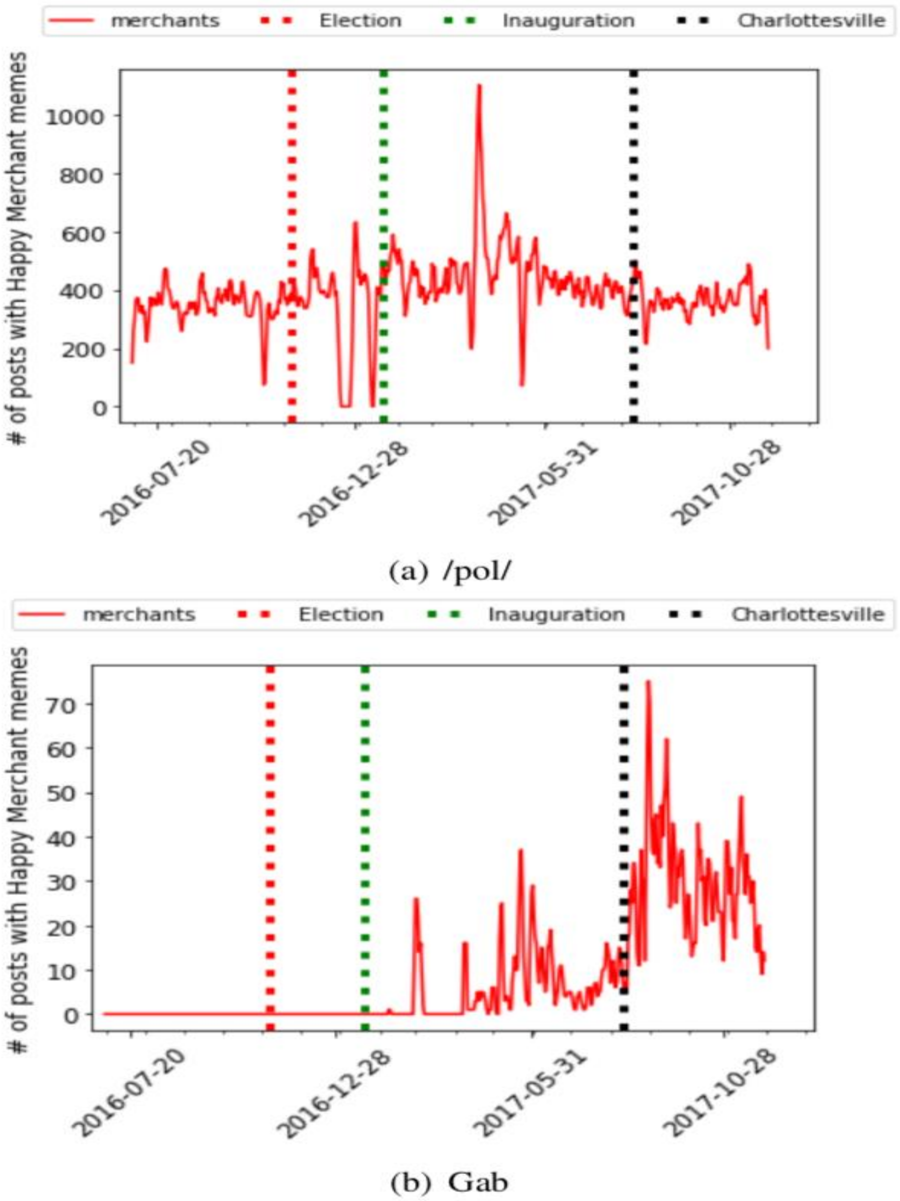


Figure 21. Graph representation of the words associated with "white" on /pol/

7.2.3. Meme analysis

In addition to hateful terms, memes also play a well-documented role in the spread of propaganda and ethnic hate in Web communities [29]. To detail how memes spread and how different Web communities influence one another with memes, our previous research [29] established a pipeline which automatically collects, annotates, and analyzes over 160M memes from over 2.6B posts from from Web communities; Reddit, /pol/, Gab, and Twitter. Within Reddit, we pay particular attention to The Donald subreddit (The Donald), a Trump supporting subreddit which notoriously propagates hateful memes [29] and propaganda [7]. In a nutshell, we use perceptual hashing [23] and clustering techniques [6] to track and analyze the propagation of memes across multiple Web communities. To achieve this, we rely on images obtained from the Know Your Meme (KYM) site [12], which is a comprehensive encyclopedia of memes.

In this work, we use this pipeline to study how antisemitic memes spread within and between these Web communities, and examine which communities are the most influential in their spread. To do this, we additionally examine two mainstream Web communities, Twitter and Reddit, and compare their influence (with respect to memes) with /pol/ and Gab. Specifically, we focus on the Happy Merchant meme [16], which is an especially important hate-meme to study in this regard for several reasons. First, it represents an unambiguous instance of antisemitic hate, and second, it is extremely popular and diverse in fringe Web communities like /pol/ and Gab [29].



(a) /pol/

(b) Gab

Figure 22. Number of posts that contain images with the Happy Merchant meme on /pol/ and Gab

First, we aim to assess the popularity and increase of use over time of the Happy Merchant meme on /pol/ and Gab. Figure 22 shows the number of posts that contain images with the Happy Merchant meme for every day of our /pol/ and Gab dataset. We further note that the numbers here represent a lower bound on the number of Happy Merchant postings: our image processing pipeline is conservative and only labels clusters that are unambiguously Happy Merchant; variations of other memes that incorporate the Happy Merchant are harder to assess. We observe that /pol/ consistently shares anti-semitic memes over time, whereas on Gab we note a substantial and sudden increase in posts containing Happy Merchant memes immediately after the Charlottesville rally. Our findings on Gab dramatically illustrate the implication that real-world eruptions of antisemitic behavior can catalyze the acceptability and popularity of antisemitic memes on other

Web communities. Taken together, these findings highlight that both communities are exploited by users to disseminate racist content that is targeted towards the Jewish community.

Another important step in examining the Happy Merchant meme is to explore how clusters of similar Happy Merchant memes relate to other meme clusters in our dataset. One possibility is that Happy Merchants make-up a unique family of memes, which would suggest that they segregate in form and shape from other memes. Given that many memes evolve from one another, a second possibility is that Happy Merchants "infect" other common memes. This could serve, for instance, to make antisemitism more accessible and common. To this end, we visualize in Figure 23 a subset of the meme clusters, which we annotate using our KYM dataset, and a Happy Merchant version of each meme. This visualization is inspired from [29] and it demonstrates numerous instances of the Happy Merchant infecting well-known and popular memes. Some examples include Pepe the Frog [17], Roll Safe [18], Bait this is Bait [13], and the Feels Good meme [14]. This suggests that users generate antisemitic variants on recognizable and popular memes.

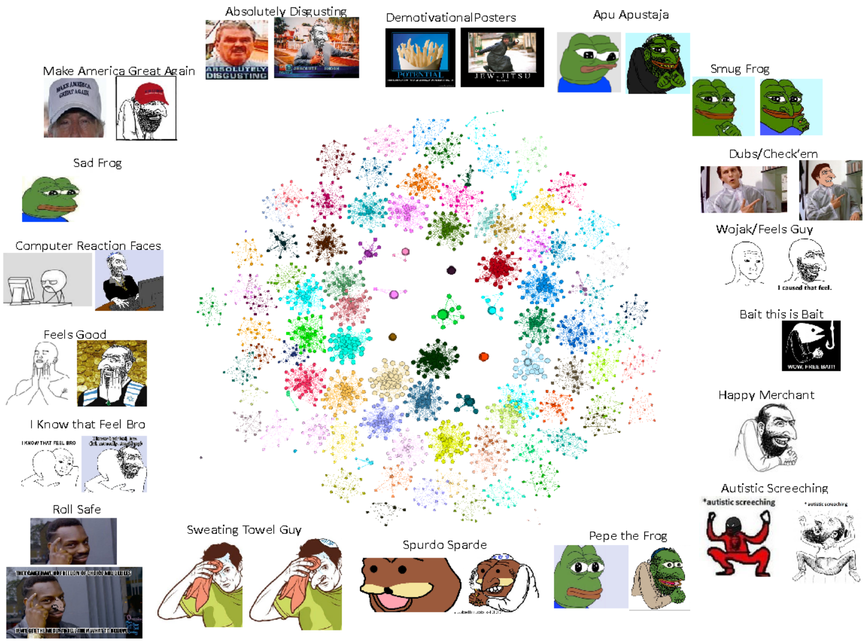


Figure 23. Visualization of a subset of the obtained image clusters with a particular focus on the penetration of the Happy Merchant meme to other seemingly neutral memes

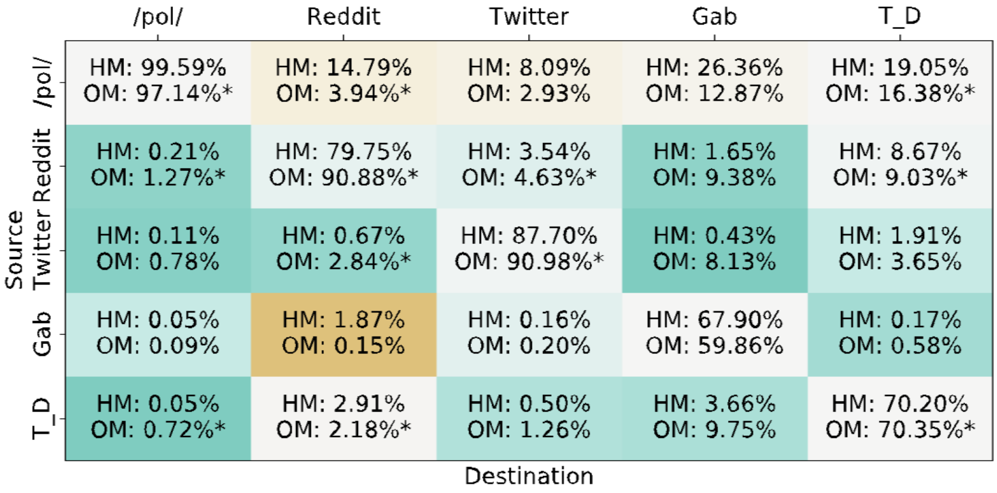
7.2.4. Influence estimation

While the growth and diversity of the Happy Merchant within fringe Web communities is a cause of significant concern, a critical question remains: How do we chart the influence of Web communities on one another in spreading the Happy Merchant? We have, until this point, examined the expanse of antisemitism on individual, fringe Web communities. Memes however, develop with the purpose to replicate and spread between different Web communities. To examine the influence of meme spread between Web communities, we employ Hawkes processes [20, 21], which can be exploited to measure the predicted, reciprocal influence that various Web communities have to each other. We fit Hawkes models for all of our annotated clusters and report the influence in two ways as in [29]. First, we report the percentage of events expected to be attributable from a source community to a destination community in Table 19. Colors indicate the percent difference between Happy

Merchants and non-Happy-Merchants, while ∗ indicate statistical significance between the distributions with p < 0.01. In other words, this shows the percentage of memes posted on one community which, in the context of our model, are expected to occur in direct response to posts in the source community. We can thus interpret this percentage in terms of the relative influence of meme postings one network on another. We also report influence in terms of efficacy by normalizing the influence that each source community has, relative to the total number of memes they post (Table 20). We compare the influence that Web communities exert on one another for the Jewish-related Happy Merchant memes (HM) and all other memes (OM) in the graph. To assess the statistical significance of the results, we perform two-sample Kolmogorov-Smirnov tests that compare the distributions of influence from the Happy Merchant and other memes; an asterisk within a cell denotes that the distributions of influence between the source and destination platform have statistically significant differences (p < 0.01).

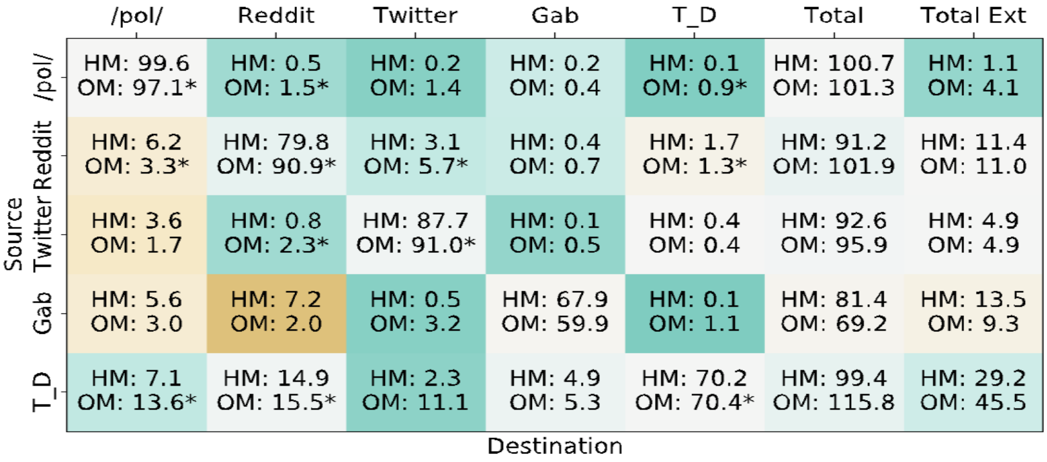
Our results show that /pol/ is the single most influential community for the spread of memes to all other Web communities. Interestingly, the influence that /pol/ exhibits in the spread of the Happy Merchant surpasses its influence in the spread of other memes. However, although /pol/’s overall influence is higher on these networks, its per-meme efficacy for the spread of antisemitic memes tended to be lower relative to non-antisemitic memes with one intriguing exception of The Donald. Another interesting feature we observe about this trend is that memes on /pol/ itself show little influence from other Web communities; both in terms of memes generally, and non-antisemitic memes in particular. This suggests a unidirectional meme flow and influence from /pol/ and furthermore, suggest that /pol/ acts as a primary reservoir to incubate and transmit antisemitism to downstream Web communities.

Table 19. Percent of the destination community’s Happy Merchant (HM) and non-Happy-Merchant (OM) meme postings caused by the source community



Source \ Destination	/pol/	Reddit	Twitter	Gab	T_D
/pol/	HM: 99.59% OM: 97.14%*	HM: 14.79% OM: 3.94%*	HM: 8.09% OM: 2.93%	HM: 26.36% OM: 12.87%	HM: 19.05% OM: 16.38%*
Reddit	HM: 0.21% OM: 1.27%*	HM: 79.75% OM: 90.88%*	HM: 3.54% OM: 4.63%*	HM: 1.65% OM: 9.38%	HM: 8.67% OM: 9.03%*
Twitter	HM: 0.11% OM: 0.78%	HM: 0.67% OM: 2.84%*	HM: 87.70% OM: 90.98%*	HM: 0.43% OM: 8.13%	HM: 1.91% OM: 3.65%
Gab	HM: 0.05% OM: 0.09%	HM: 1.87% OM: 0.15%	HM: 0.16% OM: 0.20%	HM: 67.90% OM: 59.86%	HM: 0.17% OM: 0.58%
T_D	HM: 0.05% OM: 0.72%*	HM: 2.91% OM: 2.18%*	HM: 0.50% OM: 1.26%	HM: 3.66% OM: 9.75%	HM: 70.20% OM: 70.35%*

Table 20. Influence from source to destination community of Happy Merchant and non-Happy-Merchant meme postings



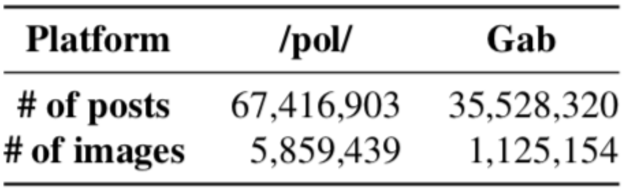
Source \ Destination	/pol/	Reddit	Twitter	Gab	T_D	Total	Total Ext
/pol/	HM: 99.6 OM: 97.1*	HM: 0.5 OM: 1.5*	HM: 0.2 OM: 1.4	HM: 0.2 OM: 0.4	HM: 0.1 OM: 0.9*	HM: 100.7 OM: 101.3	HM: 1.1 OM: 4.1
Reddit	HM: 6.2 OM: 3.3*	HM: 79.8 OM: 90.9*	HM: 3.1 OM: 5.7*	HM: 0.4 OM: 0.7	HM: 1.7 OM: 1.3*	HM: 91.2 OM: 101.9	HM: 11.4 OM: 11.0
Twitter	HM: 3.6 OM: 1.7	HM: 0.8 OM: 2.3*	HM: 87.7 OM: 91.0*	HM: 0.1 OM: 0.5	HM: 0.4 OM: 0.4	HM: 92.6 OM: 95.9	HM: 4.9 OM: 4.9
Gab	HM: 5.6 OM: 3.0	HM: 7.2 OM: 2.0	HM: 0.5 OM: 3.2	HM: 67.9 OM: 59.9	HM: 0.1 OM: 1.1	HM: 81.4 OM: 69.2	HM: 13.5 OM: 9.3
T_D	HM: 7.1 OM: 13.6*	HM: 14.9 OM: 15.5*	HM: 2.3 OM: 11.1	HM: 4.9 OM: 5.3	HM: 70.2 OM: 70.4*	HM: 99.4 OM: 115.8	HM: 29.2 OM: 45.5

7.3. Materials and Methods

Datasets

To study the extent of antisemitism on the Web, we collect two large-scale datasets from /pol/ and Gab. In this section, we shall provide a brief overview for the two communities and discuss our datasets. Table 21 summarizes the obtained datasets for both Web communities.

Table 21. Overview of our datasets. We report the number of posts and images from /pol/ and Gab



Platform	/pol/	Gab
# of posts	67,416,903	35,528,320
# of images	5,859,439	1,125,154

/pol/. 4chan is an anonymous image board that is usually exploited by troll users. A user can create a new thread by creating a post that contains an image. Other users can reply below with or without images and possibly add references to previous posts. 4chan is well-known for two features: anonymity and ephemerality. The former is the main reason that its users are more aggressive in their posts, as there is lack of accountability [3]. The latter is an interesting feature as 4chan threads usually get archived quickly (within the same day of their creation) and after one week they are permanently deleted. In this work, we focus on the Politically Incorrect board (/pol/) as it exhibits a high degree of racism and hate speech [8] and it is an influential actor on the Web’s information ecosystem [30]. To obtain data from /pol/ posts we use the same crawling infrastructure as discussed in [8], while for the images we use the methodology discussed in [29]. Specifically, we obtain posts and images posted between July 2016 and January 2018, hence acquiring 67M posts and 5.8M images.

Gab. Gab is a newly created social network, founded in August 2016, that explicitly welcomes banned users from other communities (e.g., Twitter). It waves the flag of free speech and it has mild moderation; it allows everything except illegal pornography, posts that promote terrorist acts, and doxing other users. Gab is inspired by both Twitter and Reddit in its structure. Specifically, a user can share 300-character messages with his followers (akin to Twitter), while popularity of posts within the platform is dictated via a voting system (akin to Reddit). To obtain data from Gab, we use the same methodology as described in [28] and [29] for posts and images, respectively. Overall, we obtain 35M posts and 1.1M images posted between August 2016 and January 2018.

Changepoint Analysis

To perform the changepoint analysis, we use the PELT algorithm as described in [11], and first applied to Gab timeseries data in [28]. We model each timeseries as a set of samples drawn from a normal distribution with mean and variance that are free to change at discrete times. We expect from the central limit theorem that for networks with large numbers of posts and actors, that this is a reasonable model. The algorithm then seeks to determine the points in time at which the mean and variance change by maximizing the likelihood of the distribution given the data, subject to a penalty to avoid the proliferation of changepoints. We run the algorithm with a decreasing set of penalty amplitudes. We keep track of the largest penalty amplitude at which each changepoint first appears. This gives us a ranking of the changepoints in order of their “significance.”

Hawkes Processes

To assess the root cause of the appearance of Happy Merchant memes on each of the communities, we leverage a stochastic model known as a Hawkes Process. Generally, a Hawkes model consists of K processes, where a process is a sequence of events that happen with a particular probability distribution. Colloquially, a process is analogous to a specific Web community where memes (i.e., events) are posted. Each process has a rate of events, which defines expected frequency of events on a specific Web community (for example, five posts with Happy Merchant memes per hour). An event on one process can cause impulses on other processes, which increase their rates for a period of time. An impulse is defined by a weight and a probability distribution. The former dictates the intensity of the impulse (i.e., how strong is the increase in the rate of a process), while the latter dictates how the effect of the impulse changes over time (typically it decays as time goes on). For instance, a weight of 1.5 from process A to B, means that each event on A will cause, on average, an additional 1.5 events on B.

In this work, we use a separate Hawkes model for each cluster of images that we obtained when applying the pipeline reported in [29]. Each model consists of five processes; one for each of /pol/, The Donald, the rest of Reddit, Gab, and Twitter. We elected to separate The Donald from the rest of Reddit, as it is an influential actor with respect to the dissemination of memes [29]. Next, we fit each model using Gibbs sampling as reported in [20, 21], as well as our previous research [29]. This technique enables us to obtain, at a given time, the weights and probability distributions for each impulse that is active, hence allowing us to be confident that an event is caused because of a previously occurred event on the same or on another process.

Due to the aforementioned, we argue that Hawkes Processes are a suitable framework for assessing the causal relationships between events; hence we make use of them in this work in order to quantify and understand the influence that Web communities have on each other with respect to the antisemitic Happy Merchant meme.

7.4. Discussion and Conclusion

Antisemitsm has been a historical harbinger of ethnic strife [2, 9]. While organizations have been tackling antisemitism and its associated societal issues for decades, the rise and ubiquitous nature of the Web has raised new concerns. Antisemitism and hate have grown and proliferated rapidly online and have done so mostly unchecked. This is due, in large part, to the scale and speed of the online world, and calls for new techniques to better understand and combat this worrying behavior.

In this work, we take the first step towards establishing a large-scale, scientifically grounded, quantitative understanding of antisemitism online. We analyze over 100M posts from July, 2016 to January, 2018 from two of the largest fringe communities on the Web: 4chan's Politically Incorrect board (/pol/) and Gab (a Twitter-esque service). We find evidence of increasing antisemitism and the use of racially charged language, in large part correlating with real-world political events like the 2016 US Presidential Election. We then analyze the context this language is used in via word2vec and discover several distinct facets of antisemitic language, ranging from slurs to conspiracy theories grounded in biblical literature. Finally, we examine the prevalence and propagation of the antisemitic "Happy Merchant" meme, finding that 4chan's /pol/ and Reddit's The_Donald are the most influential and efficient, respectively, in spreading this antisemitic meme across the Web.

We are certainly not the first to study antisemitism online. However, our approach differs substantially from the one traditionally taken by organizations like the Anti-Defamation League in several important ways. First, we eschew the use of surveys and qualitative analysis in favor of large-scale, data-driven, reproducible measurement. Second, our work builds upon the scientific literature resulting in well understood and open methodology. Third, the toolkit we present provides a clear direction for building automated, scalable, real-time systems to track and understand antisemitism and how it evolves over time.

That said, our work is not without limitations. First, most of our results should be considered a lower bound on the use of antisemitic language and imagery. In particular, we note that our quantification of the use of the "Happy Merchant" meme is extremely conservative. The meme processing pipeline we use is tuned in such a way that many Happy Merchant variants are clustered along with their "parent" meme. Second, our quantification of the growth antisemitic language is focused on two particular keywords, although we also show how new rhetoric is discoverable. Third, we focus primarily on two specific fringe communities. As a new community, Gab in particular is still rapidly evolving, and so treating it as a stable community (e.g., Hawkes processes), may cause us to underestimate its influence.

Regardless, there are several important recommendations we can draw from our results. First, organizations such as the ADL and SPLC should refocus their efforts towards open, data-driven methods. Small-scale, qualitative understanding is still incredibly important, especially with regard to understanding offline behavior. However, resources must be devoted to scientifically valid large-scale data analysis. More importantly, there is a need for greater transparency both in data (and its collection process) and the methods used for analysis. The scale of the problem of online hate has surpassed the ability of a single organization to solve on its own. Instead, we argue that traditional anti-hate organizations should form more intimate relationships with scientists, not just allowing, but encouraging peer-reviewed and open contributions to the scientific literature, in addition to their traditional modus operandi of public education.

Second, we believe that regardless of the participation of anti-hate organizations–scientists, and particularly computer scientists, must expend effort at understanding, measuring, and combating online antisemitism and online hate in general. The Web has changed the world in ways that were unimaginable even ten years ago. The world has shrunk, and the Information Age is in full effect. Unfortunately, many of the innovations that make the world what it is today were created with little thought to their negative consequences. For a long time, technology innovators have not considered potential negative impacts of the services they create, in some ways abdicating their responsibility to society. The present work provides solid quantified evidence that the technology that has had incredibly positive results for society is being co-opted by actors that have harnessed it in worrying ways, using the same concepts of scale, speed, and network effects to greatly expand their influence and effects on the rest of the Web and the world at large.

7.5. Section References

[1] Ego Networks. http://www.analytictech.com/networks/egonet. htm.

[2] Anti-Defamation League. Anti-Semitism, 2018.

[3] M. S. Bernstein, A. Monroy-Hernandez, D. Harry, P. Andre , K. Panovich, and G. G. Vargas. (2011). 4chan and/b: An Analysis of Anonymity and Ephemerality in a Large Online Community. In ICWSM, 2011.

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

[5] G. F. Burch, J. H. Batchelor, J. J. Burch, S. Gibson, and B. Kimball. (2018). Microaggression, anxiety, trigger warnings, emotional reasoning, mental filtering, and intellectual homogeneity on campus: A study of what students think. Journal of Education for Business, 93(5):233–241, 2018.

[6] M. Ester, H.P. Kriegel, J. Sander, X .Xu, et al. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In KDD, 1996.

[7] N. Francis. Reddit’s The Donald Was One Of The Biggest Havens For Russian Propaganda During 2016 Election, Analysis Finds. https://www.inquisitr.com/4790689/reddits-the donald-was-one-of-the-biggest-havens-for-russian-propaganda-during-2016-election-analysis-finds/, 2018.

[8] G. E. Hine, J. Onaolapo, E. De Cristofaro, N. Kourtellis, I. Leontiadis, R. Samaras, G. Stringhini, and J. Blackburn. (2017). Kek, Cucks, and God Emperor Trump: A Measurement Study of 4chan’s Politically Incorrect Forum and Its Effects on the Web. In ICWSM, 2017.

[9] History. Anti-Semitism, 2018.

[10] M. Jacomy, T. Venturini, S. Heymann, and M. Bastian. (2014). ForceAtlas2, a continuous graph layout algorithm for handy network visualization designed for the Gephi software. PloS one, 9(6):e98679, 2014.

[11] R. Killick, P. Fearnhead and A. Eckley. (2012). Optimal detection of changepoints with a linear computational cost. Journal of the American Statistical Association, 107(500):1590–1598, 2012.

[12] Know Your Meme. knowyourmeme.com/.

[13] Know Your Meme. knowyourmeme.com/memes/bait-this-is-bait, 2018.

[14] Know Your Meme. Feels Good Meme. knowyourmeme.com/memes/feels-good, 2018.

[15] Know Your Meme. Good Goy. https://knowyourmeme.com/ photos/1373391-happy-merchant, 2018.

[16] Know Your Meme. Happy Merchant Meme. knowyourmeme.com/memes/happy-merchant, 2018.

[17] Know Your Meme. Pepe the Frog Meme. knowyourmeme.com/memes/pepe-the-frog, 2018.

[18] Know Your Meme. Roll Safe Meme. http://knowyourmeme. com/memes/roll-safe, 2018.

[19] Know Your Meme The Goyim Know https://knowyourmeme. com/memes/the-goyim-know-shut-it-down, 2018.

[20] S. W. Linderman and R. P. Adams. Discovering Latent Network Structure in Point Process Data. In ICML, 2014.

[21] S. W. Linderman and R. P. Adams. Scalable Bayesian Inference for Excitatory Point Process Networks. ArXiv1507.03228, 2015.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781, 2013.

[23] V. Monga and B. L. Evans. (2016). Perceptual image hashing via feature points: performance evaluation and tradeoffs. IEEE Transactions on Image Processing, 15(11):3452–3465, 2006.

[24] S. Schama. (((SEMITISM))) Being Jewish in America in the Age of Trump, 2018.

[25] A. Thompson. The Measure of Hate on 4Chan. https://www.rollingstone.com/politics/politics-news/the-measure-of-hate-on-4chan-627922/, 2018.

[26] Urban Dictionary. Huwhite. https://www.urbandictionary.com/define.php?term=Huwhite, 2017.

[27] Wikipedia. Goy. https://en.wikipedia.org/wiki/Goy, 2018.

[28] S. Zannettou, B. Bradlyn, E. De Cristofaro, H. Kwak, M. Sirivianos, G. Stringini, and J. Blackburn. (2018). What is Gab: A Bastion of Free Speech or an Alt-Right Echo Chamber. In WWW Companion, 2018.

[29] S. Zannettou, T. Caulfield, J. Blackburn, E. De Cristofaro, M. Sirivianos, G. Stringhini, and G. Suarez-Tangil. (2018). On the Origins of Memes by Means of Fringe Web Communities. In IMC, 2018.

[30] S. Zannettou, T. Caulfield, E. De Cristofaro, N. Kourtellis, I. Leontiadis, M. Sirivianos, G. Stringhini, and J. Blackburn. (2017). The Web Centipede: Understanding How Web Communities Influence Each Other Through the Lens of Mainstream and Alternative News Sources. In IMC, 2017.

8. On the Origins of Memes by Means of Fringe Web Communities

8.1. Project description and motivation

Internet memes are increasingly used to sway and manipulate public opinion, thus prompting the need to study their propagation, evolution, and influence across the Web. In this paper, we detect and measure the propagation of memes across multiple Web communities, using a processing pipeline based on perceptual hashing and clustering techniques, and a dataset of 160M images from 2.6B posts gathered from Twitter, Reddit, 4chan’s Politically Incorrect board (/pol/), and Gab over the course of 13 months. We group the images posted on fringe Web communities (/pol/, Gab, and The_Donald subreddit) into clusters, annotate them using meme metadata obtained from Know Your Meme, and also map images from mainstream communities (Twitter and Reddit) to the clusters.

Our analysis provides an assessment of the popularity and diversity of memes in the context of each community, showing, e.g., that racist memes are extremely common in fringe Web communities. We also find a substantial number of politics-related memes on both mainstream and fringe Web communities, supporting media reports that memes might be used to enhance or harm politicians. Finally, we use Hawkes processes to model the interplay between Web communities and quantify their reciprocal influence, finding that /pol/ substantially influences the meme ecosystem with the number of memes it produces, while The_Donald has a higher success rate in pushing them to other communities.

8.2. Methodology

8.2.1. Overview

Memes are high-level concepts or ideas that spread within a culture [5]. In Internet vernacular, a meme usually refers to variants of a particular image, video, cliché, etc. that share a common theme and are disseminated by a large number of users. In this work, we focus on their most common incarnation: static images.

To gain an understanding of how memes propagate across the Web, with a particular focus on discovering the communities that are most influential in spreading them, our intuition is to build clusters of visually similar images, allowing us to track variants of a meme. We then group clusters that belong to the same meme to study and track the meme itself. In Figure 24, we provide a visual representation of the Smug Frog meme [27], which includes many variants of the same image and several clusters of variants. Cluster 1 has variants from a Jurassic Park scene, where one of the characters is hiding from two velociraptors behind a kitchen counter: the frogs are stylized to look similar to velociraptors, and the character hiding varies to express a particular message. For example, in the image in the top right corner, the two frogs are searching for an anti-semitic caricature of a Jew. Cluster N shows variants of the smug frog wearing a Nazi officer military cap with the infamous “Arbeit macht frei” in the background. Overall, these clusters represent the branching nature of memes: as a new variant of a meme becomes prevalent, it often branches into its own sub-meme, potentially incorporating imagery from other memes.

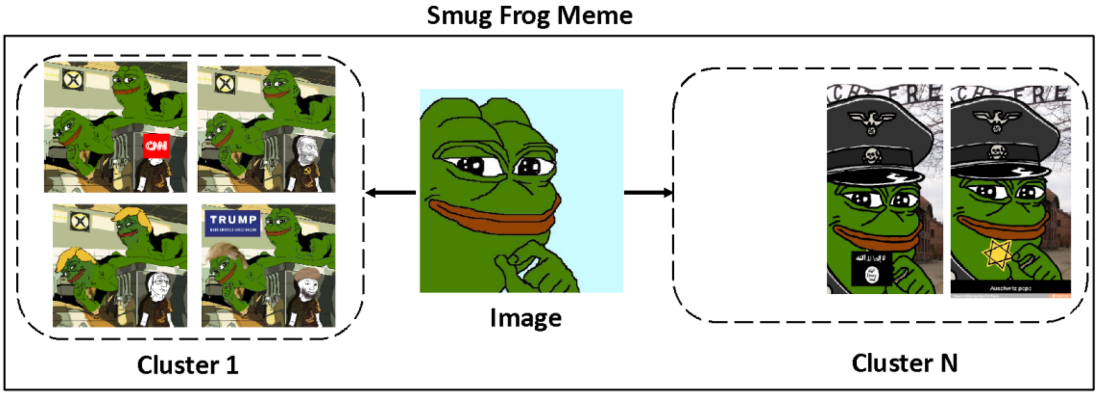


Figure 24. An example of a meme (Smug Frog) that provides an intuition of what an image, a cluster, and a meme is

We now introduce our processing pipeline, which is present in Figure 25. Our methodology aims at identifying clusters of similar images and assign them to higher level groups, which are the actual memes. Note that the proposed pipeline is not limited to image macros and can be used to identify any image. We first discuss the types of data sources needed for our approach, i.e., meme annotation sites and Web communities that post memes (dotted rounded rectangles in the Figure). Then, we describe each of the operations performed by our pipeline (Steps 1-7, see regular rectangles).

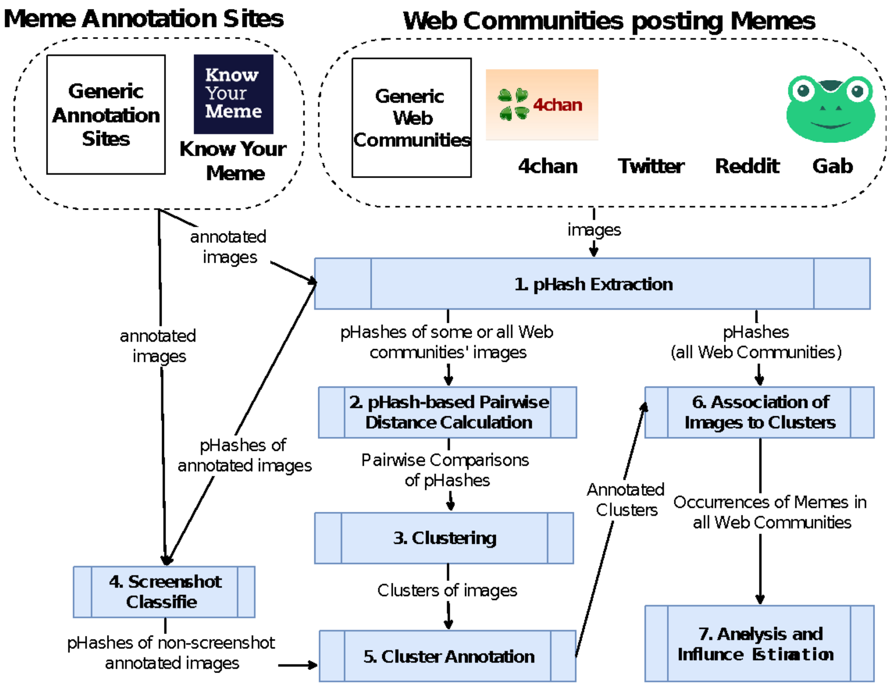

Meme Annotation Sites
Web Communities posting Memes
Generic Annotation Sites
Know Your Meme
Know Your Meme
Generic Web Communities
4chan
4chan
Twitter
Reddit
Gab
images
annotated images
annotated images
1. pHash Extraction
pHashes of some or all Web communities' images
pHashes (all Web Communities)
2. pHash-based Pairwise Distance Calculation
6. Association of Images to Clusters
pHashes of annotated images
Pairwise Comparisons of pHashes
Annotated Clusters
Occurrences of Memes in all Web Communities
3. Clustering
4. Screenshot Classifie
Clusters of images
7. Analysis and Influnce Estimation
pHashes of non-screenshot annotated images
5. Cluster Annotation

Figure 25. High-level overview of our processing pipeline

Data Sources. Our pipeline uses two types of data sources: i) sites providing meme annotation and ii) Web communities that disseminate memes. In this paper, we use Know Your Meme for the former, and Twitter, Reddit, /pol/, and Gab for the latter. However, our methodology supports any annotation site and any Web community, and this is why we add the “Generic” sites/communities notation in Figure 10.

Step 1: pHash Extraction: We use the Perceptual Hashing (pHash) algorithm [34] to calculate a fingerprint of each image in such a way that any two images that look similar to the human eye map to a “similar” hash value. pHash generates a feature vector of 64 elements that describe an image, computed from the Discrete Cosine Transform among the different frequency domains of the image. Thus, visually similar images have minor differences in their vectors. For example, the string representation of the phases obtained from the images in cluster N (see Figure 9) are 55352b0b8d8b5b53, 55952b0bb58b5353, and 55952b2b9da58a53, respectively. The algorithm is also robust against changes in the images, e.g., signal processing operations and direct manipulation [39], and effectively reduces the dimensionality of the raw images.

Steps 2 & 3 - Clustering via pairwise distance calculation: Next, we cluster images from one or more Web Communities using the pHash values. We perform a pairwise comparison of all the pHashes using Hamming distance (Step 2). To support large numbers of images, we implement a highly parallelizable system on top of TensorFlow [4], which uses multiple GPUs to enhance performance. Images are clustered using a density-based algorithm (Step 3). Our current implementation uses DBSCAN [6], mainly because it can discover clusters of arbitrary shape and performs well over large, noisy datasets. Nonetheless, our architecture can be easily tweaked to support any clustering algorithm and distance metric.

Step 4 - Screenshots Removal: Meme annotation sites like KYM often include, in their image galleries, screenshots of social network posts that are not variants of a meme but just comments about it. Hence, we discard social-network screenshots from the annotation sites data sources using a deep learning classifier.

Step 5 - Cluster Annotation: Clustering annotation uses the medoid of each cluster, i.e., the element with the minimum square average distance from all images in the cluster. In other words, the medoid is the image that best represents the cluster. The clusters’ medoids are compared with all images from meme annotation sites, by calculating the Hamming distance between each pair of pHash vectors. We consider that an image matches a cluster if the distance is less than or equal to a threshold us to capture the diversity of images that are part of the same meme while maintaining a low number of false positives. As the annotation process considers all the images of a KYM entry’s image gallery, it is likely we will get multiple annotations for a single cluster. To find the representative KYM entry for each cluster, we select the one with the largest proportion of matches of KYM images with the cluster medoid. In case of ties, we select the one with the minimum average Hamming distance. As KYM is based on community contributions it is unclear how good our annotations are. To evaluate KYM entries and our cluster annotations, three authors of this paper assessed 200 annotated clusters and 162 KYM entries. We find that only a 1.85% of the assessed KYM entries were regarded as “bad” or not sufficient. When it comes to the clustering annotation, we note that the three annotators had substantial agreement (Fleis agreement score equal to 0.67) and that the clustering accuracy, after majority agreement, of the assessed clusters is 89%.

Step 6 - Association of images to memes: To associate images posted on Web communities (e.g., Twitter, Reddit, etc.) to memes, we compare them with the clusters’ medoids, using the same threshold. This is conceptually similar to Step 5 but uses images from Web communities instead of images from annotation sites. This lets us identify memes posted in generic Web communities and collect relevant metadata from the posts (e.g., the timestamp of a tweet). Note that we track the propagation of memes in generic Web communities (e.g., Twitter) using a seed of memes obtained by clustering images from other (fringe) Web communities. More specifically, our seeds will be memes generated on three fringe Web communities (/pol/, The_Donald subreddit, Gab); nonetheless, our methodology can be applied to any community.

Step 7 - Analysis and Influence Estimation: We analyze all relevant clusters and the occurrences of memes, aiming to assess: 1) their popularity and diversity in each community; 2) their temporal evolution; 3) how communities influence each other with respect to meme dissemination.

8.3. Datasets

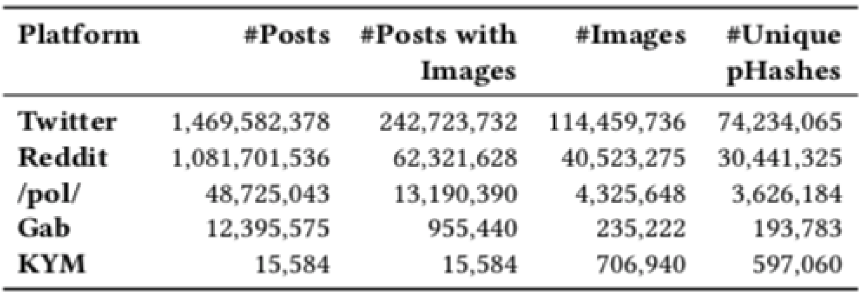
8.3.1. Web Communities

As mentioned earlier, our data sources are Web communities that post memes and meme annotation sites. For the former, we focus on four communities: Twitter, Reddit, Gab, and 4chan (more precisely, 4chan’s Politically Incorrect board, /pol/). This provides a mix of mainstream social networks (Twitter and Reddit) as well as fringe communities that are often associated with the alt-right and have an impact on the information ecosystem (Gab and /pol/) [38].

There are several other platforms playing important roles in spreading memes, however, many are “closed” (e.g., Facebook) or do not involve memes based on static images (e.g., YouTube, Giphy). In future work, we plan to extend our measurements to communities like Instagram and Tumblr, as well as to GIF and video memes. Nonetheless, we believe our data sources already allow us to elicit comprehensive insights into the meme ecosystem.

Table 22 reports the number of posts and images processed for each community. Note that the number of images is lower than the number of posts with images because of duplicate image URLs and because some images get deleted. Next, we discuss each dataset.

Table 22. Overview of our datasets



Platform	#Posts	#Posts with Images	#Images	#Unique pHashes
Twitter	1,469,582,378	242,723,732	114,459,736	74,234,065
Reddit	1,081,701,536	62,321,628	40,523,275	30,441,325
/pol/	48,725,043	13,190,390	4,325,648	3,626,184
Gab	12,395,575	955,440	235,222	193,783
KYM	15,584	15,584	706,940	597,060

Twitter. Twitter is a mainstream microblogging platform, allowing users to broadcast 280-character messages (tweets) to their followers. Our Twitter dataset is based on tweets made available via the 1% Streaming API, between July 1, 2016 and July 31, 2017. In total, we parse 1.4B tweets: 242M of them have at least one image. We extract all the images, ultimately collecting 114M images yielding 74M unique pHashes.

Reddit. Reddit is a news aggregator: users create submissions by posting a URL and others can reply in a structured way. It is divided into multiple sub-communities called subreddits, each with its own topic and moderation policy. Content popularity and ranking are determined via a voting system based on the up- and down-votes users cast. We gather images from Reddit using publicly available data from Pushshift [35]. We parse all submissions and comments1 between July 1, 2016 and July, 31 2017, and extract 62M posts that contain at least one image. We then download 40M images producing 30M unique pHashes.

4chan. 4chan is an anonymous image board; users create new threads by posting an image with some text, which others can reply to. It has two characteristic features: anonymity and ephemerality. By default, user identities are concealed, and all threads are deleted after one week. Overall, 4chan is known for its extremely lax moderation and the high degree of hate and racism, especially on boards like /pol/ [8]. We obtain all threads posted on /pol/, between July 1, 2016 and July 31, 2017, using the same methodology of [8]. Since all threads (and images) are removed after a week, we use a public archive service called 4plebs [3] to collect 4.3M images, thus yielding 3.6M unique pHashes.

Gab. Gab is a social network launched in August 2016 as a "champion" of free speech, providing "shelter" to users banned from other platforms. It combines social networking features from Twitter (broadcast of 300-character messages) and Reddit (content is ranked according to up- and down-votes). It also has extremely lax moderation as it allows everything except illegal pornography, terrorist propaganda, and doxing [36]. Overall, Gab attracts alt-right users, conspiracy theorists, and trolls, and high volumes of hate speech [37]. We collect 12M posts, posted on Gab between August 10, 2016 and July 31, 2017, and 955K posts have at least one image, using the same methodology as in [37]. Out of these, 235K images are unique, producing 193K unique pHashes.

8.3.2. Meme Annotation Site

Know Your Meme (KYM). We choose KYM as the source for meme annotation as it offers a comprehensive database of memes. KYM is a sort of encyclopedia of Internet memes: for each meme, it provides information such as its origin (i.e., the platform on which it was first observed), the year it started, as well as descriptions and examples. In addition, for each entry, KYM provides a set of keywords, called tags that describe the entry. KYM provides a variety of higher-level categories that group meme entries; namely, cultures, subcultures, people, events, and sites. "Cultures" and "sub-cultures" entries refer to a wide variety of topics ranging from video games to various general categories. For example, the Rage Comics subculture [24] is a higher-level category associated with memes related to comics like Rage Guy [25] or LOL Guy [18], while the Alt-right culture [9] gathers entries from a loosely defined segment of the right-wing community. The rest of the categories refer to specific individuals (e.g., Donald Trump [14]), specific events (e.g., #CNNBlackmail [13]), and sites (e.g., /pol/ [23]), respectively. It is also worth noting that KYM moderates all entries, hence entries that are wrong or incomplete are marked as so by the site.

As of May 2018, the site has 18.3K entries, specifically, 14K memes, 1.3K subcultures, 1.2K people, 1.3K events, and 427 websites [19]. We crawl KYM between October and December 2017, acquiring data for 15.6K entries; for each entry, we also download all the images related to it by crawling all the pages of the image gallery. In total, we collect 707K images corresponding to 597K unique pHashes. Note that we obtain 15.6K out of 18.3K entries, as we crawled the site several months before May 2018.

8.3.3. Running the pipeline in our datasets

For all four Web communities (Twitter, Reddit, /pol/, and Gab), we perform Step 1 of the pipeline, using the ImageHash library [2]. We then perform Steps 2-3 (i.e., pairwise comparisons between all images and clustering), for all the images from /pol/, The_Donald subreddit, and Gab, as we treat them as fringe Web communities. Note that, we exclude mainstream communities like the rest of Reddit and Twitter as our main goal is to obtain clusters of memes from fringe Web communities and later characterize all communities by means of the clusters. Next, we go through Steps 4-5 using all the images obtained from meme annotation websites (specifically, Know Your Meme) and the medoid of each cluster from /pol/, The_Donald, and Gab. Finally, Steps 6-7 use all the pHashes obtained from Twitter, Reddit (all subreddits), /pol/, and Gab to find posts with images matching the annotated clusters. This is an integral part of our process as it allows to characterize, and study mainstream Web communities not used for clustering (i.e., Twitter and Reddit).

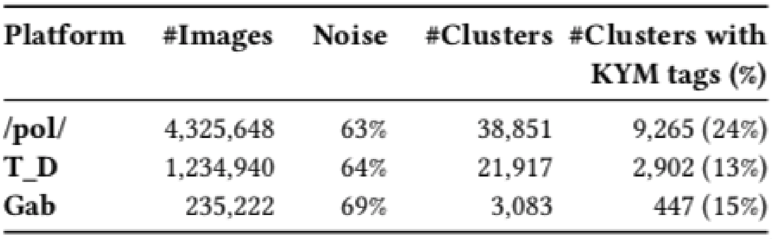
8.4. Analysis

8.4.1. Cluster-based analysis

Statistics. In Table 23, we report some basic statistics of the clusters obtained for each Web community. A relatively high percentage of images (63%–69%) are not clustered, i.e., are labeled as noise. While in DBSCAN “noise” is just an instance that does not fit in any cluster (more specifically, there are less than 5 images with perceptual distance <= 8 from that particular instance), we note that this likely happens as these images are not memes, but rather “one-off images.” For example, on /pol/ there is a large number of pictures of random people taken from various social media platforms.

Overall, we have 2.1M images in 63.9K clusters: 38K clusters for /pol/, 21K for The_Donald, and 3K for Gab. 12.6K of these clusters are successfully annotated using the KYM data: 9.2K from /pol/ (142K images), 2.9K from The_Donald (121K images), and 447 from Gab (4.5K images). As for the un-annotated clusters, manual inspection confirms that many include miscellaneous images unrelated to memes, e.g., similar screenshots of social networks posts (recall that we only filter out screenshots from the KYM image galleries), images captured from video games, etc.

Table 23. Statistics obtained from clustering images from /pol/, The_Donald, and Gab

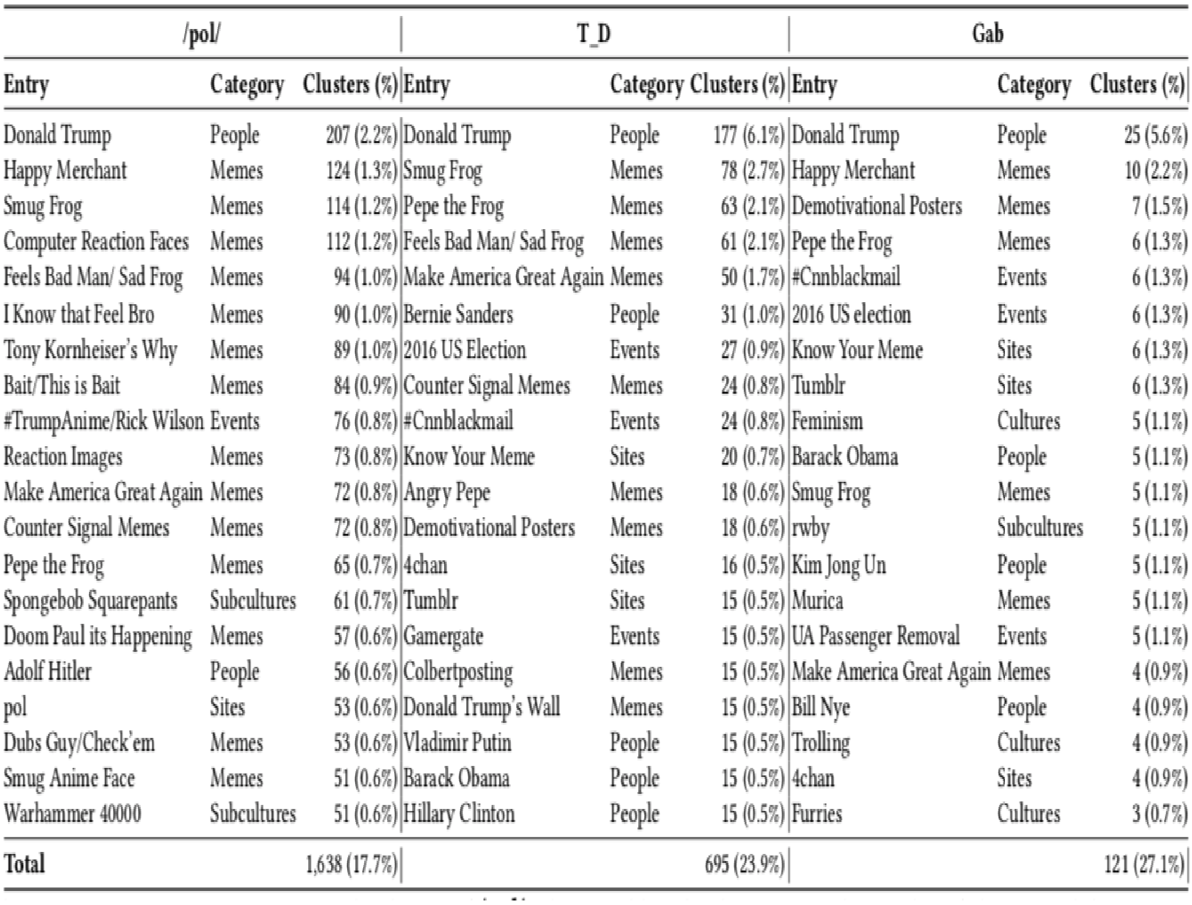


Platform	#Images	Noise	#Clusters	#Clusters with KYM tags (%)
/pol/	4,325,648	63%	38,851	9,265 (24%)
T_D	1,234,940	64%	21,917	2,902 (13%)
Gab	235,222	69%	3,083	447 (15%)

Top KYM entries. Because the majority of clusters match only one or two KYM entries, we simplify things by giving all clusters a representative annotation based on the most prevalent annotation given to the medoid, and, in the case of ties the average distance between all matches. Thus, in the rest of this report, we report our findings based on the representative annotation for each cluster.

In Table 24, we report the top 20 KYM entries with respect to the number of clusters they annotate. These cover 17%, 23%, and 27% of the clusters in /pol/, The_Donald, and Gab, respectively, hence covering a relatively good sample of our datasets. Donald Trump [14], Smug Frog [27], and Pepe the Frog [22] appear in the top 20 for all three communities, while the Happy Merchant [17] only in /pol/ and Gab. In particular, Donald Trump annotates the most clusters (207 in /pol/, 177 in The_Donald, and 25 in Gab). In fact, politics-related entries appear several times in the Table, e.g., Make America Great Again [20] as well as political personalities like Bernie Sanders, Obama, Putin, and Hillary Clinton.

Table 24. Top 20 KYM entries appearing in the clusters of /pol/, The_Donald, and Gab



/pol/			T_D			Gab		
Entry	Category	Clusters (%)	Entry	Category	Clusters (%)	Entry	Category	Clusters (%)
Donald Trump	People	207 (2.2%)	Donald Trump	People	177 (6.1%)	Donald Trump	People	25 (5.6%)
Happy Merchant	Memes	124 (1.3%)	Smug Frog	Memes	78 (2.7%)	Happy Merchant	Memes	10 (2.2%)
Smug Frog	Memes	114 (1.2%)	Pepe the Frog	Memes	63 (2.1%)	Demotivational Posters	Memes	7 (1.5%)
Computer Reaction Faces	Memes	112 (1.2%)	Feels Bad Man/ Sad Frog	Memes	61 (2.1%)	Pepe the Frog	Memes	6 (1.3%)
Feels Bad Man/ Sad Frog	Memes	94 (1.0%)	Make America Great Again	Memes	50 (1.7%)	#Cnnblackmail	Events	6 (1.3%)
I Know that Feel Bro	Memes	90 (1.0%)	Bernie Sanders	People	31 (1.0%)	2016 US election	Events	6 (1.3%)
Tony Kornheiser's Why	Memes	89 (1.0%)	2016 US Election	Events	27 (0.9%)	Know Your Meme	Sites	6 (1.3%)
Bait/This is Bait	Memes	84 (0.9%)	Counter Signal Memes	Memes	24 (0.8%)	Tumblr	Sites	6 (1.3%)
#TrumpAnime/Rick Wilson	Events	76 (0.8%)	#Cnnblackmail	Events	24 (0.8%)	Feminism	Cultures	5 (1.1%)
Reaction Images	Memes	73 (0.8%)	Know Your Meme	Sites	20 (0.7%)	Barack Obama	People	5 (1.1%)
Make America Great Again	Memes	72 (0.8%)	Angry Pepe	Memes	18 (0.6%)	Smug Frog	Memes	5 (1.1%)
Counter Signal Memes	Memes	72 (0.8%)	Demotivational Posters	Memes	18 (0.6%)	rwby	Subcultures	5 (1.1%)
Pepe the Frog	Memes	65 (0.7%)	4chan	Sites	16 (0.5%)	Kim Jong Un	People	5 (1.1%)
Spongebob Squarepants	Subcultures	61 (0.7%)	Tumblr	Sites	15 (0.5%)	Murica	Memes	5 (1.1%)
Doom Paul its Happening	Memes	57 (0.6%)	Gamergate	Events	15 (0.5%)	UA Passenger Removal	Events	5 (1.1%)
Adolf Hitler	People	56 (0.6%)	Colbertposting	Memes	15 (0.5%)	Make America Great Again	Memes	4 (0.9%)
pol	Sites	53 (0.6%)	Donald Trump's Wall	Memes	15 (0.5%)	Bill Nye	People	4 (0.9%)
Dubs Guy/Check'em	Memes	53 (0.6%)	Vladimir Putin	People	15 (0.5%)	Trolling	Cultures	4 (0.9%)
Smug Anime Face	Memes	51 (0.6%)	Barack Obama	People	15 (0.5%)	4chan	Sites	4 (0.9%)
Warhammer 40000	Subcultures	51 (0.6%)	Hillary Clinton	People	15 (0.5%)	Furries	Cultures	3 (0.7%)
Total		1,638 (17.7%)			695 (23.9%)			121 (27.1%)

When comparing the different communities, we observe the most prevalent categories are memes (6 to 14 entries in each community) and people (2-5). Moreover, in /pol/, the 2nd most popular entry, related to people, is Adolf Hilter, which supports previous reports of the community’s sympathetic views toward Nazi ideology [8]. Overall, there are several memes with hateful or disturbing content (e.g., holocaust). This happens to a lesser extent in The_Donald and Gab: the most popular people after Donald Trump are contemporary politicians.

Finally, image posting behavior in fringe Web communities is greatly influenced by real-world events. For instance, in /pol/, we find the #TrumpAnime controversy event [29], where a political individual (Rick Wilson) offended the alt-right community, Donald Trump supporters, and anime fans (an oddly intersecting set of interests of /pol/ users). Similarly, on The_Donald and Gab, we find the #Cnnblackmail [13] event, referring to the (alleged) blackmail of the Reddit user that created the infamous video of Donald Trump wrestling the CNN.

Meme Visualization. We also visualize the clusters with annotations (see Figure 26). We build a graph G = (V , E), where V are the medoids of annotated clusters and E the connections between medoids with distance under a threshold. In particular, we select this threshold as the majority of the clusters from the same meme are hierarchically connected with a higher-level cluster at a distance close to 0.45. To ease readability, we filter out nodes and edges that have a sum of in-and out-degree less than 10, which leaves 40% of the nodes and 92% of the edges. Nodes are colored according to their KYM annotation. NB: the graph is laid out using the OpenOrd algorithm [33] and the distance between the components in it does not exactly match the actual distance metric. We observe a large set of disconnected components, with each component containing nodes of primarily one color. This indicates that our distance metric is indeed capturing the peculiarities of different memes. Finally, note that an interactive version of the full graph is publicly available from [1].

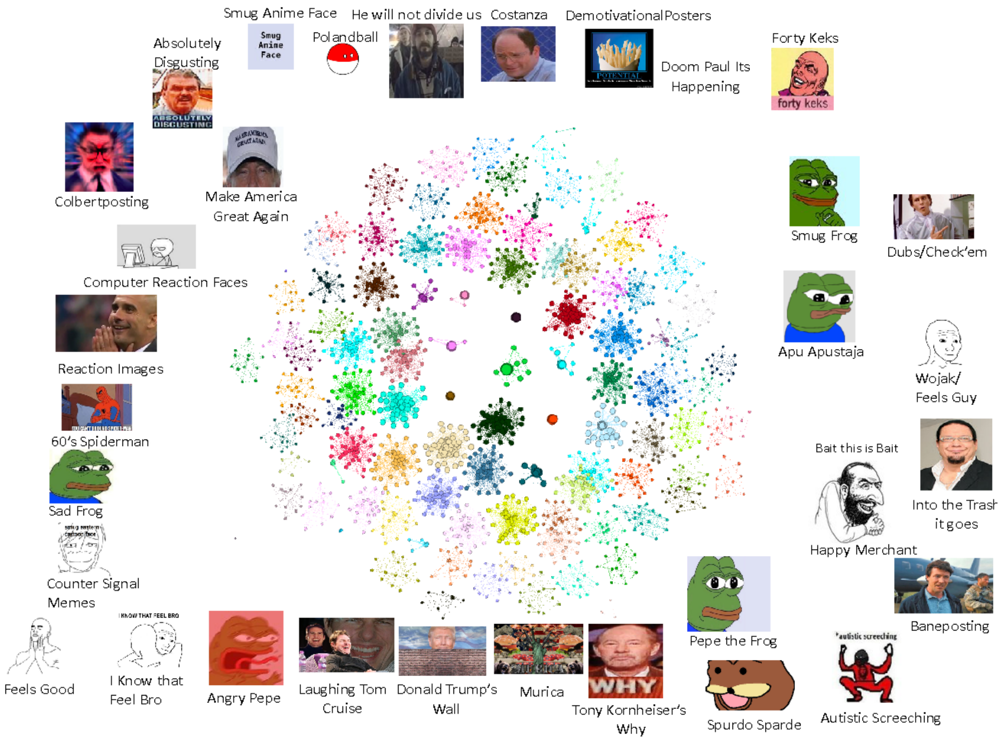


Figure 26. Visualization of the obtained clusters from /pol/, The_Donald, and Gab

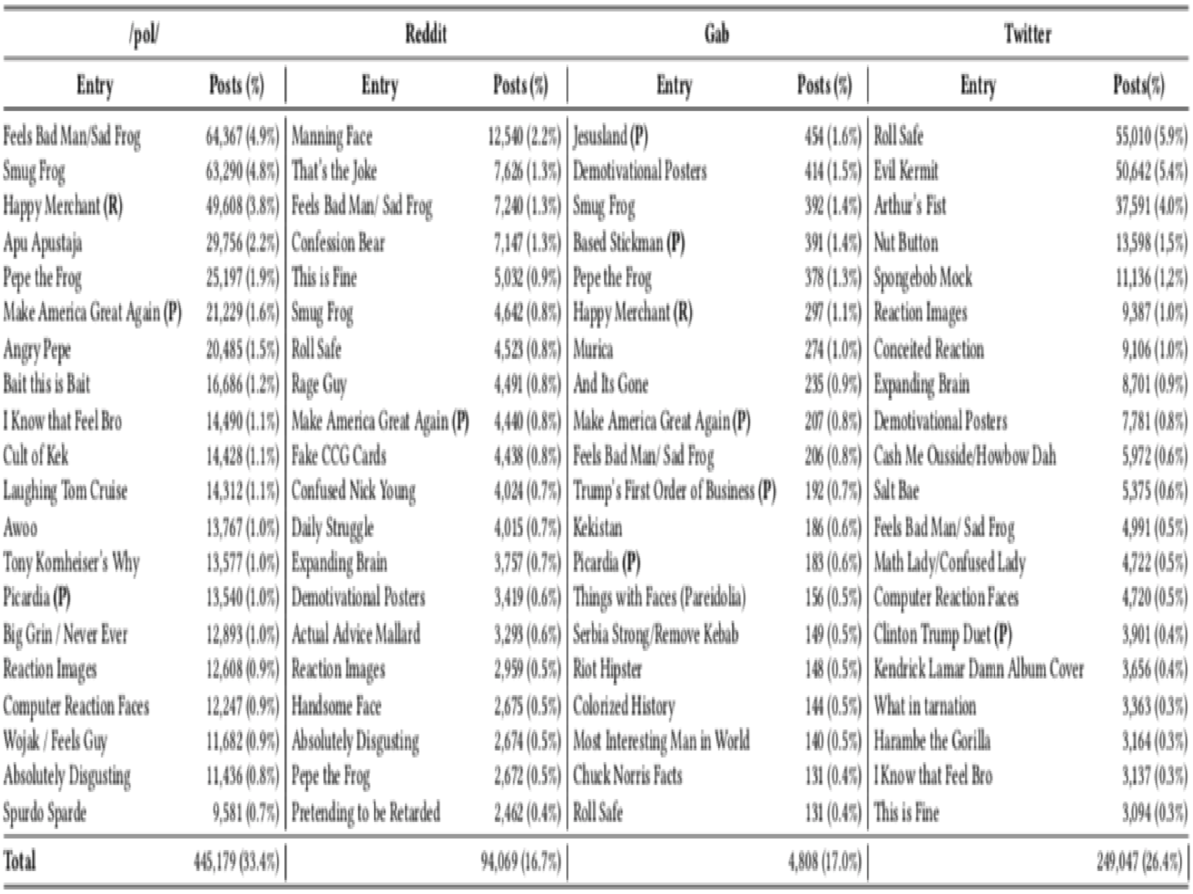
8.4.2. Web Community-based analysis

We now present a macro-perspective analysis of the Web communities through the lens of memes. We assess the presence of different memes in each community, how popular they are, and how they evolve. To this end, we examine the posts from all four communities (Twitter, Reddit, /pol/, and Gab) that contain images matching memes from fringe Web communities (/pol/, The_Donald, and Gab).

Meme Popularity. We start by analyzing clusters grouped by KYM ‘meme’ entries, looking at the number of posts for each meme in /pol/, Reddit, Gab, and Twitter.

In Table 25, we report the top 20 memes for each Web community sorted by the number of posts. We observe that Pepe the Frog [22] and its variants are among the most popular memes for every platform. While this might be an artifact of using fringe communities as a “seed” for the clustering, recall that the goal of this work is in fact to gain an understanding of how fringe communities disseminate memes and influence mainstream ones. Thus, we leave to future work a broader analysis of the wider meme ecosystem.

Table 25. Top 20 KYM entries for memes that we find in our datasets



/pol/		Reddit		Gab		Twitter	
Entry	Posts (%)	Entry	Posts (%)	Entry	Posts (%)	Entry	Posts(%)
Feels Bad Man/Sad Frog	64,367 (4.9%)	Manning Face	12,540 (2.2%)	Jesusland (P)	454 (1.6%)	Roll Safe	55,010 (5.9%)
Smug Frog	63,290 (4.8%)	That's the Joke	7,626 (1.3%)	Demotivational Posters	414 (1.5%)	Evil Kermit	50,642 (5.4%)
Happy Merchant (R)	49,608 (3.8%)	Feels Bad Man/ Sad Frog	7,240 (1.3%)	Smug Frog	392 (1.4%)	Arthur's Fist	37,591 (4.0%)
Apu Apustaja	29,756 (2.2%)	Confession Bear	7,147 (1.3%)	Based Stickman (P)	391 (1.4%)	Nut Button	13,598 (1,5%)
Pepe the Frog	25,197 (1.9%)	This is Fine	5,032 (0.9%)	Pepe the Frog	378 (1.3%)	Spongebob Mock	11,136 (1,2%)
Make America Great Again (P)	21,229 (1.6%)	Smug Frog	4,642 (0.8%)	Happy Merchant (R)	297 (1.1%)	Reaction Images	9,387 (1.0%)
Angry Pepe	20,485 (1.5%)	Roll Safe	4,523 (0.8%)	Murica	274 (1.0%)	Conceited Reaction	9,106 (1.0%)
Bait this is Bait	16,686 (1.2%)	Rage Guy	4,491 (0.8%)	And Its Gone	235 (0.9%)	Expanding Brain	8,701 (0.9%)
I Know that Feel Bro	14,490 (1.1%)	Make America Great Again (P)	4,440 (0.8%)	Make America Great Again (P)	207 (0.8%)	Demotivational Posters	7,781 (0.8%)
Cult of Kek	14,428 (1.1%)	Fake CCG Cards	4,438 (0.8%)	Feels Bad Man/ Sad Frog	206 (0.8%)	Cash Me Ousside/Howbow Dah	5,972 (0.6%)
Laughing Tom Cruise	14,312 (1.1%)	Confused Nick Young	4,024 (0.7%)	Trump's First Order of Business (P)	192 (0.7%)	Salt Bae	5,375 (0.6%)
Awoo	13,767 (1.0%)	Daily Struggle	4,015 (0.7%)	Kekistan	186 (0.6%)	Feels Bad Man/ Sad Frog	4,991 (0.5%)
Tony Kornheiser's Why	13,577 (1.0%)	Expanding Brain	3,757 (0.7%)	Picardia (P)	183 (0.6%)	Math Lady/Confused Lady	4,722 (0.5%)
Picardia (P)	13,540 (1.0%)	Demotivational Posters	3,419 (0.6%)	Things with Faces (Pareidolia)	156 (0.5%)	Computer Reaction Faces	4,720 (0.5%)
Big Grin / Never Ever	12,893 (1.0%)	Actual Advice Mallard	3,293 (0.6%)	Serbia Strong/Remove Kebab	149 (0.5%)	Clinton Trump Duet (P)	3,901 (0.4%)
Reaction Images	12,608 (0.9%)	Reaction Images	2,959 (0.5%)	Riot Hipster	148 (0.5%)	Kendrick Lamar Damn Album Cover	3,656 (0.4%)
Computer Reaction Faces	12,247 (0.9%)	Handsome Face	2,675 (0.5%)	Colorized History	144 (0.5%)	What in tarnation	3,363 (0.3%)
Wojak / Feels Guy	11,682 (0.9%)	Absolutely Disgusting	2,674 (0.5%)	Most Interesting Man in World	140 (0.5%)	Harambe the Gorilla	3,164 (0.3%)
Absolutely Disgusting	11,436 (0.8%)	Pepe the Frog	2,672 (0.5%)	Chuck Norris Facts	131 (0.4%)	I Know that Feel Bro	3,137 (0.3%)
Spurdo Sparde	9,581 (0.7%)	Pretending to be Retarded	2,462 (0.4%)	Roll Safe	131 (0.4%)	This is Fine	3,094 (0.3%)
Total	445,179 (33.4%)		94,069 (16.7%)		4,808 (17.0%)		249,047 (26.4%)

Sad Frog [16] is the most popular meme on /pol/ (4.9%), the 3rd on Reddit (1.3%), the 10th on Gab (0.8%), and the 12th on Twitter (0.5%). We also find variations like Smug Frog [27], Apu Apustaja [11], Pepe the Frog [22], and Angry Pepe [10]. Considering that Pepe is treated as a hate symbol by the Anti-Defamation League [30] and that is often used in hateful or racist, this likely indicates that polarized communities like /pol/ and Gab do use memes to incite hateful conversation. This is also evident from the popularity of the anti-semitic Happy Merchant meme [17], which depicts a “greedy” and “manipulative” stereotypical caricature of a Jew (3.8% on /pol/ and 1.1% on Gab).

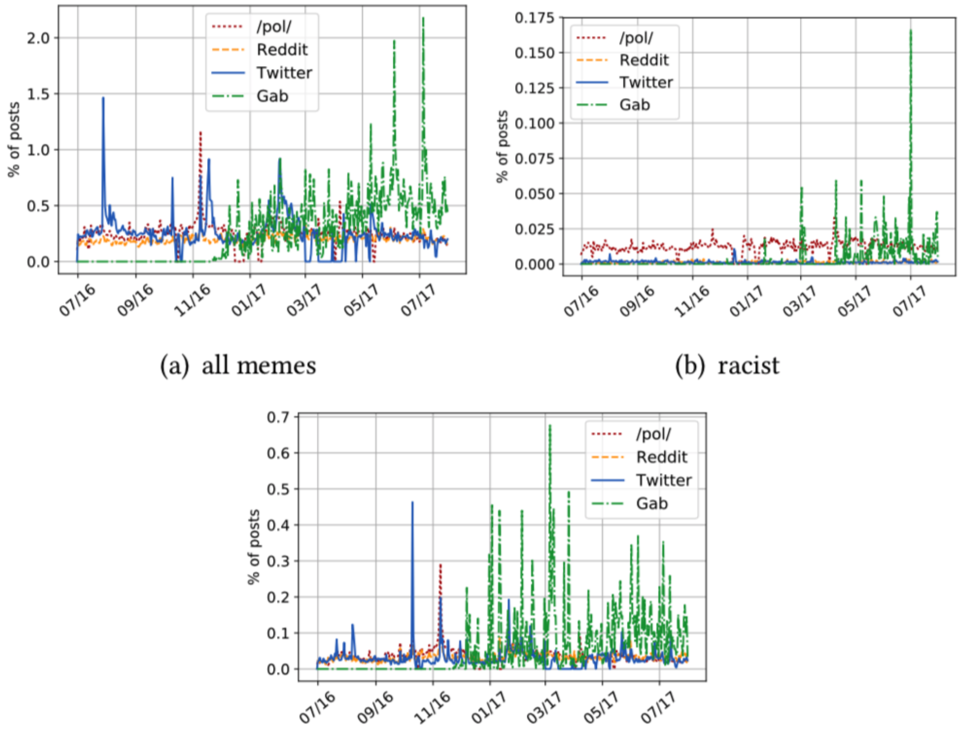
By contrast, mainstream communities like Reddit and Twitter primarily share harmless/neutral memes, which are rarely used in hateful contexts. Specifically, on Reddit the top memes are Manning Face [21] (2.2%) and That’s the Joke [28] (1.3%), while on Twitter the top ones are Roll Safe [26] (5.9%) and Evil Kermit [15] (5.4%).

Once again, we find that users (in all communities) post memes to share politics-related information, possibly aiming to enhance or penalize the public image of politicians. For instance, we find Make America Great Again [20], a meme dedicated to Donald Trump’s US presidential campaign, among the top memes in /pol/ (1.6%), in Reddit (0.8%), and Gab (0.8%). Similarly, in Twitter, we find the Clinton Trump Duet meme [12] (0.4%), a meme inspired by the 2nd US presidential debate.

We further group memes into two high-level groups, racist and politics-related. We use the tags that are available in our KYM dataset, i.e., we assign a meme to the politics-related group if it has the “politics,” “2016 us presidential election,” “presidential election,” “trump,” or “clinton” tags, and to the racism-related one if the tags include “racism,” “racist,” or “antisemitism,” obtaining 117 racist memes(4.4% of all memes that appear on our dataset) and 556 politics-related memes (21.2% of all memes that appear on our dataset). In the rest of this report we use these groups for our analysis.

Temporal Analysis. Next, we study the temporal aspects of posts that contain memes from /pol/, Reddit, Twitter, and Gab. In Figure 27, we plot the percentage of posts per day that include memes. For all memes, we observe that /pol/ and Reddit follow a steady posting behavior, with a peak in activity around the 2016 US elections. We also find that memes are increasingly more used on Gab (see, e.g., 2016 vs 2017).

Both /pol/ and Gab include a substantially higher number of posts with racist memes, used over time with a difference in behavior: while /pol/ users share them in a very steady and constant way, Gab exhibits a bursty behavior. A possible explanation is that the former is inherently more racist, with the latter primarily reacting to particular world events. As for political memes, we find a lot of activity overall on Twitter, Reddit, and /pol/, but with different spikes in time. On Reddit and /pol/, the peaks coincide with the 2016 US elections. On Twitter, we note a peak that coincides with the 2nd US Presidential Debate on October 2016. For Gab, there is again an increase in posts with political memes after January 2017.



(a) all memes

(b) racist

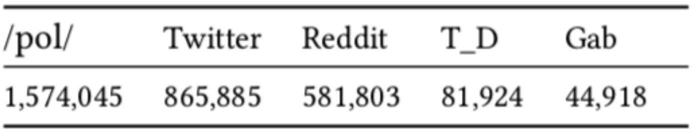
(c) politics

Figure 27. Percentage of posts per day in our dataset for all, racist, and politics-related memes

8.5. Influence Estimation

We fit Hawkes models using Gibbs sampling as described in [32] for the 12.6K annotated clusters; in Table 26, we report the total number of meme images posted to each community in these clusters. We note that /pol/ has the greatest number of memes posted, followed by Twitter and Reddit. Recall, however, that because our approach is seeded with memes observed on /pol/, The_Donald, and Gab, it is possible that there are memes on Twitter and Reddit that are not included in the clusters. In addition, the raw number of images (not necessarily memes) that appear on the different communities varies greatly. This yields an additional interesting question: how efficient are different communities at disseminating memes?

Table 26. Events per community from the 12.6K clusters

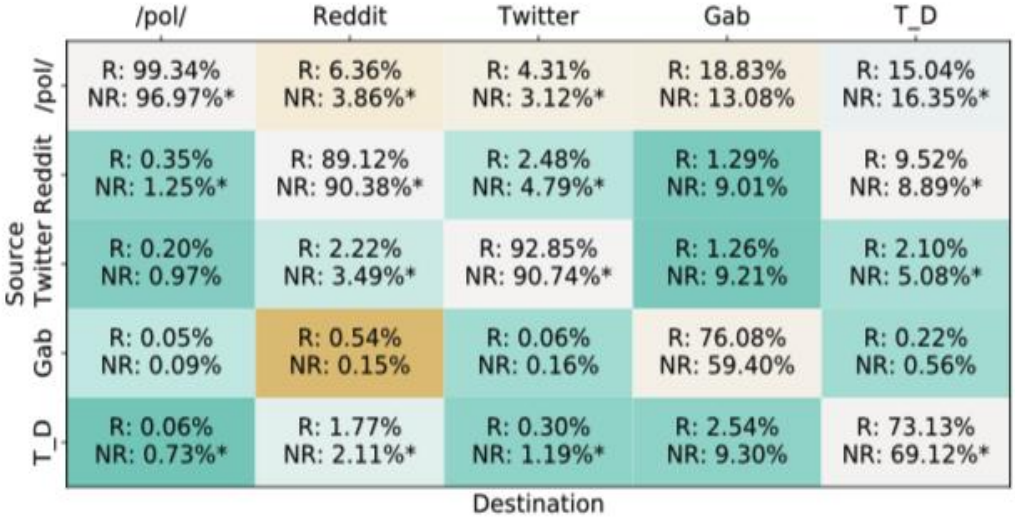


/pol/	Twitter	Reddit	T_D	Gab
1,574,045	865,885	581,803	81,924	44,918

First, we report the source of events in terms of the percent of events on the destination community. This describes the results in terms of the data as we have collected it, e.g., it tells us the percentage of memes posted on Twitter that were caused by /pol/. The second way we report influence is by normalizing the values by the total number of events in the source community, which lets us see how much influence each community has, relative to the number of memes they post—in other words, their efficiency.

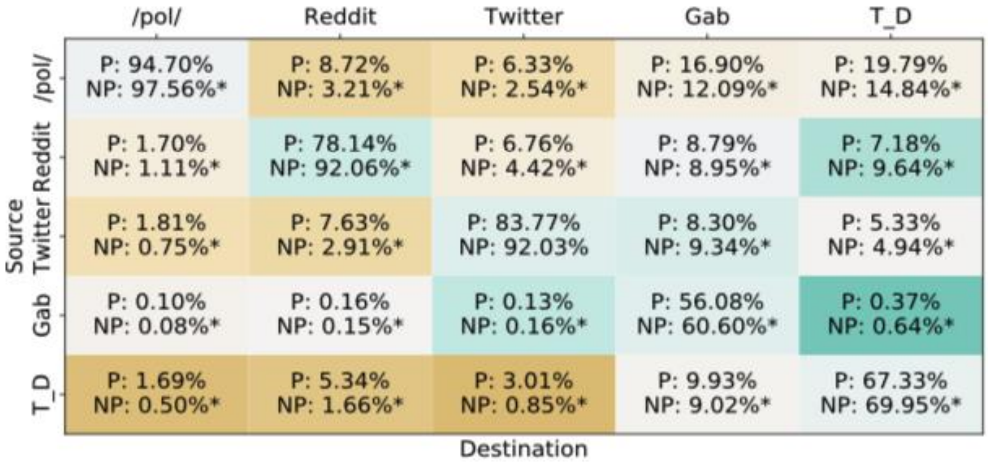
Using the clusters identified as either racist or non-racist, we compare how the communities influence the spread of these two types of content. Table 27 shows the percentage of both the destination community’s racist and non-racist meme posts caused by the source community. Colors indicate the percent difference between racist and non-racist. We perform two-sample Kolmogorov-Smirnov tests to compare the distributions of influence from the racist and non-racist clusters; cells with statistically significant differences between influence of racist/non-racist memes (with p<0.01) are reported with a * in the Table 28. /pol/ has the most total influence for both racist and non-racist memes, with the notable exception of Twitter, where Reddit has the most the influence. Interestingly, while the percentage of racist meme posts caused by /pol/ is greater than non-racist for Reddit, Twitter, and Gab, this is not the case for The_Donald. The only other cases where influence is greater for racist memes are Reddit to The_Donald and Gab to Reddit.

Table 27. Percent of the destination community's racist (R) and non-racist (NR) meme postings caused by the source community


Source
Destination

	/pol/	Reddit	Twitter	Gab	T_D
/pol/	R: 99.34% NR: 96.97%*	R: 6.36% NR: 3.86%*	R: 4.31% NR: 3.12%*	R: 18.83% NR: 13.08%	R: 15.04% NR: 16.35%*
Reddit	R: 0.35% NR: 1.25%*	R: 89.12% NR: 90.38%*	R: 2.48% NR: 4.79%*	R: 1.29% NR: 9.01%	R: 9.52% NR: 8.89%*
Twitter	R: 0.20% NR: 0.97%	R: 2.22% NR: 3.49%*	R: 92.85% NR: 90.74%*	R: 1.26% NR: 9.21%	R: 2.10% NR: 5.08%*
Gab	R: 0.05% NR: 0.09%	R: 0.54% NR: 0.15%	R: 0.06% NR: 0.16%	R: 76.08% NR: 59.40%	R: 0.22% NR: 0.56%
T_D	R: 0.06% NR: 0.73%*	R: 1.77% NR: 2.11%*	R: 0.30% NR: 1.19%*	R: 2.54% NR: 9.30%	R: 73.13% NR: 69.12%*

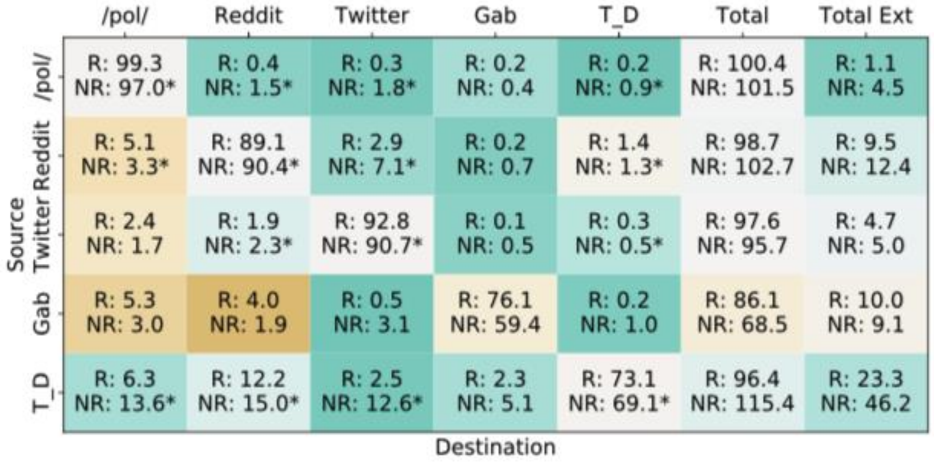
Table 28. Percent of the destination community's political (P) and non-political (NP) meme postings caused by the source community


Source
Destination

	/pol/	Reddit	Twitter	Gab	T_D
/pol/	P: 94.70% NP: 97.56%*	P: 8.72% NP: 3.21%*	P: 6.33% NP: 2.54%*	P: 16.90% NP: 12.09%*	P: 19.79% NP: 14.84%*
Reddit	P: 1.70% NP: 1.11%*	P: 78.14% NP: 92.06%*	P: 6.76% NP: 4.42%*	P: 8.79% NP: 8.95%*	P: 7.18% NP: 9.64%*
Twitter	P: 1.81% NP: 0.75%*	P: 7.63% NP: 2.91%*	P: 83.77% NP: 92.03%	P: 8.30% NP: 9.34%*	P: 5.33% NP: 4.94%*
Gab	P: 0.10% NP: 0.08%*	P: 0.16% NP: 0.15%*	P: 0.13% NP: 0.16%*	P: 56.08% NP: 60.60%*	P: 0.37% NP: 0.64%*
T_D	P: 1.69% NP: 0.50%*	P: 5.34% NP: 1.66%*	P: 3.01% NP: 0.85%*	P: 9.93% NP: 9.02%*	P: 67.33% NP: 69.95%*

When looking at political vs non-political memes (Table 29), we see a somewhat different story. Here, /pol/ influences The_Donald more in terms of political memes. Further, we see differences in the percent increase and decrease of influence between the Table 29 and Table 30 (as indicated by the cell colors). For example, Twitter has a relatively larger difference in its influence on /pol/ and Reddit for political and non-political memes than for racist and non-racist memes, but a smaller difference in its influence on Gab and The_Donald. This exposes how different communities have varying levels of influence depending on the type of memes they post.

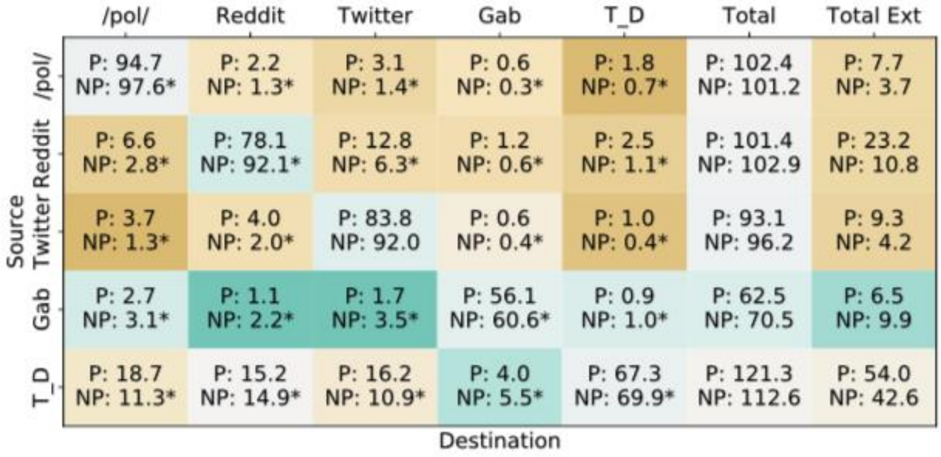
Table 29. Influence from source to destination community of racist and non-racist meme postings, normalized by the number of events in the source community



Source	/pol/	Reddit	Twitter	Gab	T_D	Total	Total Ext
/pol/	R: 99.3 NR: 97.0*	R: 0.4 NR: 1.5*	R: 0.3 NR: 1.8*	R: 0.2 NR: 0.4	R: 0.2 NR: 0.9*	R: 100.4 NR: 101.5	R: 1.1 NR: 4.5
Reddit	R: 5.1 NR: 3.3*	R: 89.1 NR: 90.4*	R: 2.9 NR: 7.1*	R: 0.2 NR: 0.7	R: 1.4 NR: 1.3*	R: 98.7 NR: 102.7	R: 9.5 NR: 12.4
Twitter	R: 2.4 NR: 1.7	R: 1.9 NR: 2.3*	R: 92.8 NR: 90.7*	R: 0.1 NR: 0.5	R: 0.3 NR: 0.5*	R: 97.6 NR: 95.7	R: 4.7 NR: 5.0
Gab	R: 5.3 NR: 3.0	R: 4.0 NR: 1.9	R: 0.5 NR: 3.1	R: 76.1 NR: 59.4	R: 0.2 NR: 1.0	R: 86.1 NR: 68.5	R: 10.0 NR: 9.1
T_D	R: 6.3 NR: 13.6*	R: 12.2 NR: 15.0*	R: 2.5 NR: 12.6*	R: 2.3 NR: 5.1	R: 73.1 NR: 69.1*	R: 96.4 NR: 115.4	R: 23.3 NR: 46.2

Destination

Table 30. Influence from source to destination community of political and non-political meme postings, normalized by the number of events in the source community



Source	/pol/	Reddit	Twitter	Gab	T_D	Total	Total Ext
/pol/	P: 94.7 NP: 97.6*	P: 2.2 NP: 1.3*	P: 3.1 NP: 1.4*	P: 0.6 NP: 0.3*	P: 1.8 NP: 0.7*	P: 102.4 NP: 101.2	P: 7.7 NP: 3.7
Reddit	P: 6.6 NP: 2.8*	P: 78.1 NP: 92.1*	P: 12.8 NP: 6.3*	P: 1.2 NP: 0.6*	P: 2.5 NP: 1.1*	P: 101.4 NP: 102.9	P: 23.2 NP: 10.8
Twitter	P: 3.7 NP: 1.3*	P: 4.0 NP: 2.0*	P: 83.8 NP: 92.0	P: 0.6 NP: 0.4*	P: 1.0 NP: 0.4*	P: 93.1 NP: 96.2	P: 9.3 NP: 4.2
Gab	P: 2.7 NP: 3.1*	P: 1.1 NP: 2.2*	P: 1.7 NP: 3.5*	P: 56.1 NP: 60.6*	P: 0.9 NP: 1.0*	P: 62.5 NP: 70.5	P: 6.5 NP: 9.9
T_D	P: 18.7 NP: 11.3*	P: 15.2 NP: 14.9*	P: 16.2 NP: 10.9*	P: 4.0 NP: 5.5*	P: 67.3 NP: 69.9*	P: 121.3 NP: 112.6	P: 54.0 NP: 42.6

Destination

While examining the raw influence provides insights into the meme ecosystem, it obscures notable differences in the meme posting behavior of the different communities. To explore this, we look at the normalized influence in Tables 29 (racist/non-racist memes) and 30 (political/non-political memes). As mentioned previously, normalization reveals how efficient the communities are in *disseminating* memes to other communities by revealing the per meme influence of meme posts. First, we note that the percent change in influence for the dissemination of racist/non-racist memes is quite a bit larger than that for political/non-political memes (again, indicated by the coloring of the cells). More interestingly, both Figures show that, contrary to the total influence, /pol/ is the least influential when taking into account the number of memes posted. While this might seem surprising, it actually yields a subtle, yet crucial aspect of /pol/’s role in the meme ecosystem: /pol/ (and 4chan in general) acts as an evolutionary microcosm for memes. The constant production of new content [8] results in a “survival of the fittest” [7] scenario. A staggering number of memes are posted on /pol/, but only the best actually makes it out to other communities. To the best of our knowledge, this is the first result quantifying this analogy to evolutionary pressure.

8.6. Conclusion

In this work, we presented a large-scale measurement study of the meme ecosystem. We introduced a novel image processing pipeline and ran it over 160M images collected from four Web communities (4chan’s /pol/, Reddit, Twitter, and Gab). We clustered images from fringe communities (/pol/, Gab, and Reddit’s The_Donald) based on perceptual hashing and a custom distance metric, annotated the clusters using data gathered from Know Your Meme, and analyzed them along a variety of axes. We then associated images from all the communities to the clusters to characterize them through the lens of memes and the influence they have on each other.

Our analysis highlights that the meme ecosystem is quite complex, with intricate relationships between different memes and their variants. We found important differences between the memes posted on different communities (e.g., Reddit and Twitter tend to post “fun” memes, while Gab and /pol/ racist or political ones). When measuring the influence of each community toward disseminating memes to other Web communities, we found that /pol/ has the largest overall influence for racist and political memes, however, /pol/ was the least efficient, i.e., in terms of influence w.r.t. the total number of memes posted, while The_Donald is very successful in pushing memes to both fringe and mainstream Web communities.

Our work constitutes the first attempt to provide a multi-platform measurement of the meme ecosystem, with a focus on fringe and potentially dangerous communities. Considering the increasing relevance of digital information on world events, our study provides a building block for future cultural anthropology work, as well as for building systems to protect against the dissemination of harmful ideologies. Moreover, our pipeline can already be used by social network providers to assist the identification of hateful content; for instance, Facebook is taking steps to ban Pepe the Frog used in the context of hate [31], and our methodology can help them automatically identify hateful variants. Finally, our pipeline can be used for tracking the propagation of images from any context or other language spheres, provided an appropriate annotation dataset.

8.7. Section References

[1] Graph visualization of the clusters. https://memespaper.github.io/.

[2] ImageHash Python Library. https://github.com/JohannesBuchner/imagehash.

[3] 4plebs. 4chan archive. http://4plebs.org/.

[4] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al. TensorFlow: A System for Large-Scale Machine Learning. In OSDI, 2016.

[5] R. Dawkins. The selfish gene. Oxford university press, 1976.

[6] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In KDD, 1996.

[7] C. Faife. How 4Chan’s Structure Creates a ’Survival of the Fittest’ for Memes. https://motherboard.vice.com/en_us/article/ywzm8m/how-4chans-structure-creates-a-survival-of-the-fittest-for-memes, 2017.

[8] G. E. Hine, J. Onaolapo, E. De Cristofaro, N. Kourtellis, I. Leontiadis, R. Samaras, G. Stringhini, and J. Blackburn. Kek, Cucks, and God Emperor Trump: A Measurement Study of 4chan’s Politically Incorrect Forum and Its Effects on the Web. In ICWSM, 2017.

[9] Know Your Meme. Alt-Right Culture. http://knowyourmeme.com/memes/cultures/alt-right, 2018.

[10] Know Your Meme. Angry Pepe Meme. http://knowyourmeme.com/memes/angry-pepe, 2018.

[11] Know Your Meme. Apu Apustaja Meme. http://knowyourmeme.com/memes/apu-apustaja, 2018.

[12] Know Your Meme. Clinton/Trump Duet Meme. http://knowyourmeme.com/memes/make-america-great-again, 2018.

[13] Know Your Meme. #CNNBlackmail. http://knowyourmeme.com/memes/events/cnnblackmail, 2018.

[14] Know Your Meme. Donald Trump Meme. http://knowyourmeme.com/memes/people/donald-trump, 2018.

[15] Know Your Meme. Evil Kermit Meme. http://knowyourmeme.com/memes/evil-kermit, 2018.

[16] Know Your Meme. Feels Bad Man / Sad Frog Meme. http://knowyourmeme.com/memes/feels-bad-man-sad-frog, 2018.

[17] Know Your Meme. Happy Merchant Meme. http://knowyourmeme.com/memes/happy-merchant, 2018.

[18] Know Your Meme. LOL Guy Meme. http://knowyourmeme.com/memes/lol-guy, 2018.

[19] Know Your Meme. Main page of KYM entries. http://knowyourmeme.com/memes/, 2018.

[20] Know Your Meme. Make America Great Again Meme. http://knowyourmeme.com/memes/make-america-great-again, 2018.

[21] Know Your Meme. Manning Face Meme. http://knowyourmeme.com/memes/manningface, 2018.

[22] Know Your Meme. Pepe the Frog Meme. http://knowyourmeme.com/memes/pepe-the-frog, 2018.

[23] Know Your Meme. /pol/ KYM entry. http://knowyourmeme.com/memes/sites/pol,2018.

[24] Know Your Meme. Rage Comics Subculture. http://knowyourmeme.com/memes/subcultures/rage-comics, 2018.

[25] Know Your Meme. Rage Guy Meme. http://knowyourmeme.com/memes/age-guy-fffuuuuuuuu, 2018.

[26] Know Your Meme. Roll Safe Meme http://knowyourmeme.com/memes/roll-safe,2018.

[27] Know Your Meme. Smug Frog Meme. http://knowyourmeme.com/memes/smug-frog, 2018.

[28] Know Your Meme. That’s The Joke Meme. http://knowyourmeme.com/memes/thats-the-joke, 2018.

[29] Know Your Meme. #TrumpAnime / Rick Wilson Controversy. http://knowyourmeme.com/memes/events/trumpanime-rick-wilson-controversy, 2018.

[30] A.-D. League. Pepe the Frog. https://www.adl.org/education/references/hate-symbols/pepe-the-frog, 2018.

[31] S. Lerner. Facebook To Ban Pepe The Frog Images Used In The Context Of Hate. http://www.techtimes.com/articles/228632/20180525/facebook-publishes-official-policy-on-pepe-the-frog.htm, 2018.

[32] S. W. Linderman and R. P. Adams. Scalable Bayesian Inference for Excitatory Point Process Networks. ArXiv 1507.03228, 2015.

[33] S. Martin, W. M. Brown, R. Klavans, and K. W. Boyack. OpenOrd: An Open-source Toolbox for Large Graph Layout. In Visualization and Data Analysis 2011, 2011.

[34] V. Monga and B. L. Evans. Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs. IEEE Transactions on Image Processing, 2006.

[35] Pushshift. Reddit Data. http://files.pushshift.io/reddit/, 2018.

[36] P. Snyder, P. Doerfler, C. Kanich, and D. McCoy. Fifteen minutes of unwanted fame: Detecting and characterizing doxing. In IMC, 2017.

[37] S. Zannettou, B. Bradlyn, E. De Cristofaro, M. Sirivianos, G. Stringhini, H. Kwak,and J. Blackburn. What is Gab? A Bastion of Free Speech or an Alt-Right Echo Chamber? In WWW Companion, 2018.

[38] S. Zannettou, T. Caulfield, E. De Cristofaro, N. Kourtellis, I. Leontiadis, M. Sirivianos, G. Stringhini, and J. Blackburn. The Web Centipede: Understanding How Web Communities Influence Each Other Through the Lens of Mainstream and Alternative News Sources. In IMC, 2017.

[39] C. Zauner, M. Steinebach, and E. Hermann. Rihamark: perceptual image hash benchmarking. In Media Forensics and Security, 2011.

9. Summary and Future Work

In this document we provided details about the ongoing work related to the development of automated techniques to detect early cyberbullying patterns through emotional analysis, online abuse, online antisemitism detection and how hateful memes originating from fringe communities reach and affect mainstream online social networks.

Most of the above mentioned work is deployed in the ENCASE Framework. The efforts listed in this document helped the project reach a big milestone with regards to identifying online abuse and how to protect minors from it. The ENCASE Framework, equipped with the aforementioned techniques will be tested and be piloted during WP7.

All the projects above have made significant steps towards automatically detecting malicious behavior and the project reached a very important milestone with this document.

10. Publications

The following publications have been submitted as part of our effort in D4.2.

1. "***On the Origins of Memes by Means of Fringe Web Communities***". Savvas Zannettou, Tristan Caulfield, Jeremy Blackburn, Emiliano De Cristofaro, Michael Sirivianos, Gianluca Stringhini, and Guillermo Suarez-Tangil. ACM Internet Measurement Conference (IMC), 2018.
2. “***A Quantitative Approach to Understanding Online Antisemitism***”. Joel Finkelstein, Savvas Zannettou, Barry Bradlyn, Jeremy Blackburn. Arxiv, 2018.

11. Copyright and Intellectual Property

The intellectual property will be jointly owned between the Institutions that each of the ENCASE partners. If a project partner decides to move institutions for the duration of the project the Institution to which they move would not become a join owner, and the ownership will remain with the institution at which partners are originally based.