“You Know What to Do”: Proactive Detection of YouTube Videos Targeted by Coordinated Hate Attacks

Enrico Mariconti[1], Guillermo Suarez-Tangil[1,2], Jeremy Blackburn[3], Emiliano De Cristofaro[1], Nicolas Kourtellis[4], Ilias Leontiadis[4], Jordi Luque Serrano[4], Gianluca Stringhini[1]

[1]University College London, [2]Kings College London, [3]University of Alabama at Birmingham, [4]Telefonica

Abstract

Over the years, the Web has shrunk the world, allowing individuals to share viewpoints with many more people than they are able to in real life. At the same time, however, it has also enabled anti-social and toxic behavior to occur at an unprecedented scale. Video sharing platforms like YouTube receive uploads from millions of users, covering a wide variety of topics and allowing others to comment and interact in response. Unfortunately, these communities are periodically plagued with aggression and hate attacks. In particular, recent work has showed how these attacks often take place as a result of *“raids,”* i.e., organized efforts coordinated by ad-hoc mobs from third-party communities.

Despite the increasing relevance of this phenomenon, online services often lack effective countermeasures to mitigate it. Unlike well-studied problems like spam and phishing, coordinated aggressive behavior both targets and is perpetrated by *humans*, making defense mechanisms that look for automated activity unsuitable. Therefore, the de-facto solution is to *reactively* rely on user reports and human reviews. In this paper, we propose an automated solution to identify videos that are likely to be targeted by coordinated harassers. First, we characterize and model YouTube videos along several axes (metadata, audio transcripts, thumbnails) based on a ground truth dataset of raid victims. Then, we use an ensemble of classifiers to determine the likelihood that a video will be raided with high accuracy (AUC up to 94%). Overall, our work paves the way for providing video platforms like YouTube with *proactive* systems to detect and mitigate coordinated hate attacks.

1 Introduction

As social interactions increasingly take place on the Web, cyber-aggression has unfortunately become a pressing problem [28, 57]. In particular, coordinated harassment campaigns are more and more frequent, with perpetrators working together to deliver harmful content in a repetitive fashion [11, 12, 19]. One example of such behavior is a phenomenon known as *raiding*, i.e., **ad-hoc mobs coordinating on social platforms to organize and orchestrate attacks aimed to disrupt other platforms and undermine users** who advocate for issues and policies they do not agree with [30, 37].

Nonetheless, service providers are often criticized for providing inadequate countermeasures [62, 63]. Abusive activity is generated by humans and not by automated programs, thus, systems used to detect unwanted content, bots, etc. [6, 5, 10, 45, 59, 60, 66] are not easily adapted to this problem. Hence, platforms mostly adopt *reactive* solutions, letting users report abusive accounts and taking actions according to terms of services, e.g., blocking or suspending offenders [35]. However, this approach is inherently slow, and limited by biases in the reports and by the resources available to verify them. Moreover, this inevitably yields an arms race with the abusers, who can create new accounts when they get blocked.

In this paper, we focus on **raids against YouTube videos**. We do so since: (1) YouTube is one of the top visited sites worldwide, with more than 1 billion users and 1 billion hours of videos watched every day [70], and (2) it is targeted by aggressive behavior and extremism [44], as recently acknowledged by Google [8].

Moreover, prior work [30] shows that YouTube is the most heavily targeted platform by hateful and alt-right communities within 4chan, and in particular the Politically Incorrect board (/pol/). 4chan-led raids typically start with a user posting a link to a YouTube video on /pol/, often with comments like *“you know what to do,”* resulting into a spike in the number of hateful comments on the YouTube video. The authors of [30] also examine the degree of synchronous commenting behavior between 4chan and YouTube, validating it in terms of the rate of hate comments on the video page, as well as commenter account overlap. However, besides providing a characterization that identifies when a raid has occurred, previous work has not proposed any solutions to mitigate the problem.

In this paper, we propose a *proactive* approach towards curbing coordinated hate attacks against YouTube users. Rather than looking at attacks as they happen, or at known abusive accounts, **we investigate whether we can automatically identify YouTube videos that are likely to be raided.** We present a system that relies on multiple features of YouTube videos, such as title, category, thumbnail preview, as well as audio transcripts, to build a model of the characteristics of videos that are commonly raided. This also allows us to gain an understanding of what content attracts raids, i.e., *why* these videos are raided. We use a ground truth dataset of 428 raided YouTube videos obtained from [30], comparing them to 15K regular YouTube videos that were not targeted by raiders. Based on our analysis, we then build classification models to assess, at upload time, whether a video is likely to be raided in the future. We actually rely on an *ensemble* of classifiers, each looking at a different element of the video (metadata, thumbnails, and audio transcripts), and build an ensemble detection algorithm that performs quite well, reaching AUC values of up to 94%.

Our work is an important first step towards curbing raids on video sharing platforms, as we shows that proactive measures can work to detect videos targeted by coordinated hate attacks. Providers like YouTube could integrate these techniques in multiple configurations: for instance, they could analyze every video that gets uploaded to the platform and take particular precautions for the ones that are flagged as in danger of being raided (e.g., vetting or rate-limiting the comments that they receive). Alternatively, they could monitor links to YouTube videos posted on communities that are know to organize raids against other platforms (e.g., 4chan's /pol/), automatically learning which of these videos are actually likely to be targeted, and then take similar precautions.

In summary, our paper makes the following contributions:

- We analyze and model YouTube raids, perpetrated by users of 4chan, using a ground truth dataset of 428 raided videos;
- We build an ensemble classification model geared to determine the likelihood that a YouTube video will be raided in the future, using a number of features (video metadata, audio transcripts, thumbnails). Our system achieves an AUC of 94% when analyzing raided videos posted on 4chan with respect to all other non raided videos in our dataset.
- We provide concrete suggestions as to how video platforms can deploy our methodology to detect raids and mitigate their impact.

Figure 1: Example of comments from raided YouTube videos, with usernames and profile pictures removed for the sake of privacy.

2 Background & Datasets

Hate attacks on online services can happen in a number of ways. In this paper, we focus on organized attacks – *"raids"* – which are orchestrated by a community and target users of other platforms [30, 37]. In this section, we provide an overview of online raids, and describe how fringe communities organize and orchestrate them. We then detail the datasets collected for our experiments.

2.1 Anatomy of Online Raids

Unlike "typical" attacks on online services like DDoS [55], a raid is an attack on the *community* that calls a service home. The goal is not to disrupt the service itself, but rather to cause chaos and turmoil to the *users* of the service. As such, online raids are a growing *socio-technical* problem. Nonetheless, it is hard to provide a precise definition of them. In the following, we offer a description of them based on previous work as well as our own observations of raids in the wild.

A prototypical raid begins with a user finding a YouTube video and posting a link to it on a 3rd party community, e.g. 4chan. In some cases, the original poster, or another user, might also write comments like "you know what to do." Shortly after, the YouTube video starts receiving a large number of negative and hateful comments. Overall, raids present a couple of key characteristics. For instance, they typically attract a large number of users, joining an effort to explicitly disrupt any productive/civil discourse that might be occurring. This is different from what would normally happen with possibly controversial videos (say, e.g., a video posted by social justice advocates), which also attract opposing points of view, though organically. The raid often generates a sudden, unexpected attack by otherwise uninvolved users.

Consider, for example, the comments from a few raided videos showed in Figure 1. The first comment specifically calls out the racist ideology espoused on 4chan (and based on the comparative analysis between different 4chan boards from [30], this commenter is likely referring to /pol/ in particular). Another user—presumably a 4chan user—responds by telling the commenter to kill themselves. The next set of comments refers to the racist meme that Italians are not "white." The first response also uses the anti-semitic triple parenthesis meme [68] to imply that the TED organization (the source of the video being commented on) is a tool of a supposed Jewish conspiracy. When another user responds that Italians are in fact white, the original commenter provides a justification for

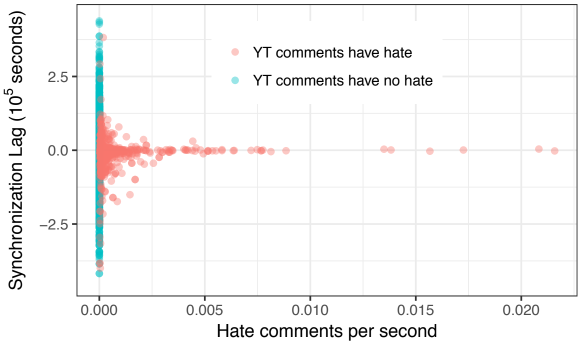


Figure 2: Distribution of videos from [30], according to the synchronization of their comments with the 4chan thread where the URL was posted and the number of hate words that appear in the comments.

his assertion: that Italians are too Mediterranean to be considered white. In the final set of comments, a user asks why the "raiders" are even watching the video if they have issues with the poster's opinions; the response is that they need to ensure that the video's poster (a minority) "knows his place."

Another characteristics of raids is their semi-coordinated nature. While a sudden increase in hateful comments to a video is obvious to an outside observer, what is not obvious is the fact that these comments are part of a coordinated attack. In fact, those participating in a raid may even discuss the "fruits of their labor" on the 3rd party site that they organize on. For example, as discussed in [30], /pol/ threads serve as an aggregation point for raiders; users will post a hateful comment to the targeted YouTube video, and then brag about it on /pol/. This observation led the authors to identify videos that might have been targeted by a raid by measuring the number of "hate comments per second" (HCPS) and the synchronization between the comments posted on the YouTube video and those appearing on the /pol/ thread advocating for a raid.

By playing with the synchronization lag and the HPCS metric, the authors of [30] proceeded to identify a set of videos that had received raids during their observation period. This approach was further validated by showing an increase in the overlap of YouTube accounts between videos as the synchronization lag decreases: the same accounts were more likely to be involved in the YouTube comments. In other words, it was not random YouTube users leaving racist comments, but rather "serial" raiders almost assuredly originating from /pol/. In Figure 2, we show the distribution of the videos in the dataset by [30] according to synchronization lag and HCPS: observe that, the closer the lag is to zero, the higher rate of hate comments received by the video.

2.2 Datasets

To better understand online raids against YouTube videos, and develop a proactive detection system, we first need to collect real world data. To this end, we use three datasets, also summarized in Table 1:

1. We collect a set of videos that were raided after being posted on /pol/, as identified by previous work [30];

Type	Source	# Videos
Raided	4chan (/pol/)	428
Non-Raided	4chan (/pol/)	789
Random	YouTube	14,444

Table 1: Overview of our dataset of YouTube videos. Source denotes the place where the link to the YouTube video was collected.

2. We gather a set of videos that were posted on /pol/ which were *not* raided;

3. We retrieve a set of random YouTube videos, which we use to compare raided videos against, aiming to highlight the differences in their characteristics.

Raided videos posted on 4chan (ground truth). We start by collecting a ground truth dataset of raided YouTube videos. As discussed above, fringe communities within 4chan are often responsible for organizing raids against YouTube users that promote ideas that they do not agree with. Therefore, we obtain the dataset of YouTube links posted on 4chan over a 2.5-month period in 2016 (June to mid September) from the authors of [30]. For our purposes, we want to choose *conservative* thresholds to ensure we only select videos that we are confident have been raided. Thus, based on Figure 2, we select videos with $HCPS > 10^{-4}$ and time lag less than a day, resulting in 428 videos (out of 1,217) that were raided. We manually examined this ground truth dataset to further increase our confidence that they were indeed raided.

Non-raided videos posted on 4chan. Although many YouTube videos posted on 4chan's /pol/ are victims of raids, obviously not all videos posted attract hateful behavior. Figure 2 shows that videos that have a high lag compared to the thread in which they are posted are unlikely to see much hateful behavior. To compare the characteristics of these videos to the raided ones, we build a second dataset with videos that were posted on 4chan but were *not* raided. We use conservative thresholds to ensure that we do not mistakenly included raided videos: to be part of this set, a video needs to have both a synchronization lag of more than one day compared to the 4chan thread it was posted in, and to have a HCPS of 0. Ultimately, this yields 789 non-raided videos.

Random YouTube videos. Finally, in order to draw comparisons with the ground truth of raided videos, we need to collect a set of YouTube videos that are likely not raided. We use the YouTube API and download 50 of the top videos across a variety of categories. In the end, we collected 14,444 videos, selected following the same distribution of (YouTube) categories as those linked on 4chan.

Ethical considerations. Our data collection received ethics approval from our institution (details are conceived to preserve this submission's anonymity). We also took a few steps to follow standard ethical practices, e.g., discarded *any* personal information about the users uploading or commenting on the videos, encrypted data at rest, etc.

3 Video Processing and Analysis

We now present the methods used to analyze the characteristics of the YouTube videos in our dataset that received raids. We look at the metadata of a video, its audio transcript, as well as the thumbnail preview. We then use the insights derived from this analysis to build a machine learning classifier geared to determine whether a YouTube video is likely to receive a raid (see Section 5).

3.1 Metadata

In addition to the actual videos, we also collect the associated metadata, specifically: title, duration, category, description, and tags. Except for the duration, these fields are entered by the user uploading the video, so they might not always be meaningful or might even be missing. Naturally, title, duration, and description often play a major role in a user's decision to watch the video as they are the first fields that they see. Also, the tags provide an intuition of a video's topics, and are actually also used by YouTube to suggest other videos—in a way, watching a suggested video might actually trigger a post on 4chan. Looking at the category for videos posted on 4chan, we anecdotally find that many of them include news, politics, and ethnic issues.

Evidence of controversial topics. We perform term frequency-inverse document frequency (*TF-IDF*) analysis on the string metadata (title, tags, and description) to extract information about the most used keywords in the different groups of videos, finding that random videos often include "Google," "music," and "love" (top 3 used words), as well as "follow" and "subscribe." By contrast, all videos posted on 4chan include politics-related words such as "Hillary" and "Trump," or indications of racial content like "black", while only raided videos have certain words like "police," "lives" (likely related to the Black Lives Matter movement), or "Alex" (referring to InfoWars' Alex Jones, a conspiracy theorist supporting of Trump, who is well known in alt-right circles).[1]

The differences in the topics used in the metadata are extremely important: search engines are affected by the content of the metadata, especially tags; moreover YouTube suggests videos to the user based on many of these fields. Overall, we observe that random YouTube videos have few topics in common with the 4chan videos, while there are some similarities between the set of videos posted on 4chan but not raided and those that have been raided.

3.2 Audio Processing

The process to extract audio from each video involve five steps. (1) First, we download YouTube videos in MPEG-4 format, with H.264 video and AAC audio codecs, then, (2) we extract the corresponding stereo audio channels using the ffmpeg tool at 44.1KHz sampling rate. (3) Both audio channels are then mixed and down-sampled to 8KHz, using the sox utility; this operation is necessary to have the same conditions between the YouTube audios and the samples used to train the Voice Activity Detection (VAD) system used in the following step. (4) Next, we rely on a VAD module to discriminate non-speech audio segments for further processing, and finally, (5) we use an Automatic Speech Recognition (ASR) system, based on deep neural networks and trained with conversational telephone data, to perform the speech-to-text transcription.

Voice Activity Detection. VAD is often used as an upstream processing step intended to prevent unwanted data from entering later stages. The VAD system we use is based on [17] and uses long short-term memory (LSTM) recurrent neural networks. We train it using 20 hours of call center data annotated with speech and non-speech labels. The audio waveform is parametrized by using 12 Mel frequency cepstral coefficients (MFCC) at 10ms frame rate. The system is evaluated on 1.4 hours of call center data, with error rates ranging from 5% to 8% with 20 and 10 input frames, respectively.

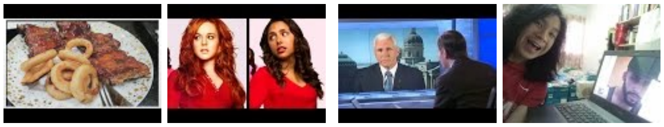
Automatic Speech Recognition. We use an ASR system for English from [41], trained using the Kaldi toolkit [53] and the Switchboard corpus [24], which includes around 300 hours of conversational speech. In particular, we adapt the Switchboard training recipe for nnet2 models from [53], and train two different systems. The first uses a GMM/HMM, specifically, a triphone unit with a decision tree of 5,500 leafs and a total of 90,000 Gaussian mixtures, The second re-uses the first model but switches Gaussian mixtures with a vanilla DNN with 4 hidden layers and 1,024 neurons per layer. The GMM system makes use of "discriminative" feature transformations for GMM alignment.

For the language modeling, we estimate a trigram language model using MIT Language Model Toolkit with Kneser-Ney Smoothing [31]. It is worth mentioning that we do not perform lattice re-scoring with any other language model. The pronunciation dictionary, an orthographic/phonetic mapping, is from CMUdict, an open source pronunciation dictionary.[2] The target lexicon accounts for more than 40K words. Note that neither "bad words" nor slang terms are in the original Switchboard lexicon. To evaluate the ASR performance, we use a separated subset of the same Switchboard database accounting for 5 hours of speech. The results obtained by the DNN based system, trained using only the Switchboard dataset, show a 13.05% Word Error Rate (WER). We also run the speech recognition system on the videos dataset and employ segmental Minimum Bayes-Risk decoding (sMBR) [25] to generate the best decoding transcription, also known as the one-best transcription.

Evidence of controversial topics. Similar to what done with the metadata, we also analyze the transcribed words to compare the different datasets. We observe that most YouTube videos have a lot of verbal communication. Specifically, 86% of the videos have at least 10 words spoken with the median and average video transcription containing 317 and 1,200 words respectively. We also look at whether or not some terms are more prevalent in raided YouTube videos, by averaging the *TF-IDF* vectors separately for the two classes (raided and

[1] https://en.wikipedia.org/wiki/Alex_Jones

[2] http://www.speech.cs.cmu.edu/cgi-bin/cmudict



(a) Non-raided. **(b)** Non-raided. **(c)** Raided. **(d)** Raided.

Figure 3: Sample of thumbnails from our dataset.

Type	Non-Raid	Raid	Diff
Clothing	25.5%	33.4%	7.9%
Male-Gender	52.4%	59.1%	6.7%
Device	44.3%	50.7%	6.4%
Vehicle	8.9%	12.4%	3.4%
Animal	9.2%	5.8%	3.4%
Sport	22.6%	20.3%	2.2%
Color	12.5%	10.7%	1.8%
Joy	1.6%	2.8%	1.2%
Culture	1.6%	0.7%	0.9%
Food	2.45%	1.6%	0.8%
Female-Gender	9.8%	10.3%	0.5%
Nature	6.8%	6.8%	0.02%

Table 2: Topics found across videos with thumbnails.

non-raided videos), and examining the most influential terms. We find words like "black," "police," "white," "shot," "world," "gun," "war," "American," "government," and "law" in the top 20 terms in raided videos (in addition to some stop words and extremely popular terms that were excluded). Of these, the only word that appears among the top 20 in the non-raided videos is "government." The top terms for non-raided videos are different: they include words like "god," "fun," "movie," and "love."

3.3 Thumbnails

On YouTube, each video is also represented by an image thumbnail, used, e.g., in search results. Thumbnails provide viewers with a quick snapshot of the content of each video. Although users can manually select them [69], by default, thumbnails are automatically selected from the video and the user can choose one of three suggested options.

Using the YouTube API, we extract all available thumbnails from the videos in our dataset—specifically, using a combination of image recognition and content extraction tools (see below). Note that out of 2,895 videos, 430 (276 non-raided and 154 raided) thumbnails are not available. In a few cases this happens because the videos were not accessible via the API when we attempted to collect the thumbnails (which was done separately from the video itself, comments, and metadata), but, most of the time, the thumbnail has not been uploaded properly and were therefore inaccessible even though the video was still available.

Image recognition. To extract meaningful information from the thumbnails, we use deep neural networks [34, 65]. A large corpus of images can be used to train a deep neural network: each image is annotated with visible context that is used as ground truth, and the resulting network can then recognize objects appearing in the images and generate an accurate description of them.

We build on the work by Vinyals et al. [65] to train a generative model on top of a deep Neural Network (NN), more specifically a convolutional NN and a language-generating recurrent NN. The model is built from a very extensive dataset of annotated images (over 300,000) collected by Microsoft's COCO (Common Objects in Context) project.[3] The system outputs a detailed description of the image provided as input.

Context extraction. For each thumbnail, we then output a description that represents the semantics involved in the image. Figure 3 shows images from four examples of different thumbnails, two in the *raided* category and two in the *non-raided* category. The following descriptions have been automatically inferred from each of the images: (a) a white plate topped with a pile of food, (b) a couple of women standing next to each other, (c) a man in a suit and tie standing in front of a TV, and (d) a woman sitting in front of a laptop computer.

Note that each caption extracted not only identifies the main actor within the picture (a plate, a couple of women, or a man), but also the background activity. However, these descriptions are automatically inferred based on a model bounded by the objects in the training images-set, thus, there might be misinterpretations. Nonetheless, we believe that the 300K image COCO dataset is large and diverse enough to meet our needs.

Evidence of controversial topics. We use *topic-modeling* to abstract the descriptions obtained from the images using ConcepNet [56]. We extract categories related to *sports*, *joy*, *underage*, *gender*, to name a few. For example, some of the words in the *joy* category are "happy," "smile," "wedding," or "Christmas." Table 2 shows a summary of the prevalence of words related to any of these categories across videos in our dataset. We observe that there are a number of common topics displayed across both classes (e.g., *nature*). Likewise, female gender references are almost equal in both classes, with a slight bias towards *raid* videos. Interestingly, the case of male gender references is clearly biased towards *raided* videos, with males appearing in about 52% of the non-raided videos and in 59% of the raided ones. Reference to clothes (e.g., "tie", "dress", "jacket", "uniform") is the most distinctive category with a 7.9% difference between each class.

This indicates that there are a number of thumbnails whose context can be used to characterize videos that could potentially be raided. However, numbers also indicate that thumbnails alone might not be able to fully model the differences between the two classes. Our intuition at this point is that thumbnails can contribute towards the classification decisions, but they will not outperform other feature sets.

4 Proactive Detection

In this section, we introduce our approach to provide a *proactive* detection tool for videos targeted by hate attacks on online services, and on YouTube in particular. Our goal is to systematize this task, using supervised learning—specifically, relying on a set of machine learning classifiers, each of which focuses

[3] http://mscoco.org/

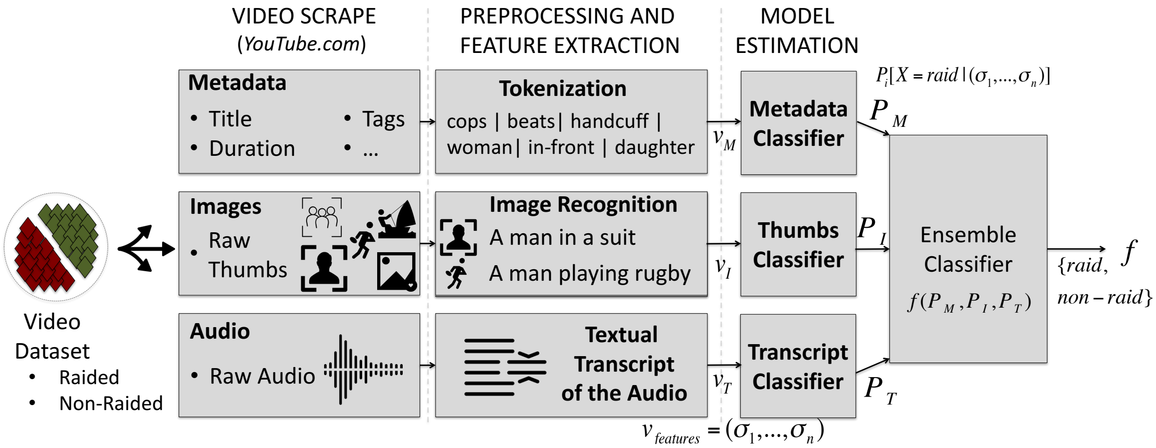


Figure 4: Architecture of our proactive detection system.

on a different set of features extracted from online videos.

In this section, we describe the set of features we use, motivated by the findings reported in the previous section. Since not all the videos contain the same information, we present three independent classifiers, one for each feature set, then show how to ensemble and properly balance the different predictions to provide one single decision.

4.1 Overview

A high-level description of our detection system is presented in Figure 4. The system is first trained using a dataset of *raided* and *non-raided* videos from different sources (we describe the different settings in Section 5.1). We do so to obtain the key elements discussed next, which will be used to predict whether a video could be targeted by hate attacks or not.

(1) A set of prediction models $\mathcal{C} = \{C_1, \ldots, C_i\}$ that output the probability $C_i(\sigma_1, \ldots, \sigma_n) = P_i[Y = \texttt{raid} \mid (\sigma_1, \ldots, \sigma_n)]$ of each video Y being raided given a feature vector $(\sigma_1, \ldots, \sigma_n)$ obtained from different elements i of the video. Each of these prediction models are referred to as *individual classifiers*.

(2) A weighted model $f(\mathcal{C}) = \sum w_i \cdot C_i$ that combines all predictions in $\mathcal{C}$, where each of the classifiers C_i is weighted by w_i based on the performance obtained on a validation set. This set is different from the training set (used to build the individual probabilities) and the testing set (used to measure the efficacy of the classifiers). The process of weighting the individual predictors also serves as a way to calibrate the output of the probabilities. The final classifier will then output a decision based on a voting such that

$$f = \begin{cases} \texttt{raid} & \text{if } f(\mathcal{C}) < \epsilon \\ \texttt{non-raid} & \text{otherwise,} \end{cases}$$

where ϵ is a threshold typically set to $\left\lfloor \frac{\sum w_i}{2} \right\rfloor + 1$.

To ease presentation, we refer to the model presented in (2) as *weighted-vote*. One can simplify the model by giving equal weight to all w_i (typically $w_i = 1$) and obtaining a nominal value for C_i before voting. In other words, applying a threshold for each C_i (e.g., 0.5) and creating an equal vote among participants. We refer to this non-weighted voting system as *majority-vote*. One can further simplify the scheme by combining each individual prediction using the arithmetic mean of the output the probabilities—this is known as an *average-prediction* system. More details about the each of the classifiers are presented later on—in particular, Section 4.3.1 and Section 4.3.2 present the architecture of the individual classifiers, and Section 4.3.3 the architecture of the ensemble used in our operational setting.

Note that the parameters (i.e., w_i, ϵ, and the thresholds for deciding the class in each C_i) used in both *majority-vote* and *average-prediction* are fixed and do not require calibration. Thus, the validation set is not used in these two modes.

4.2 Feature Engineering

In the following, we discuss how we create the features vectors used by the different classifiers.

Our system extracts features from three different sources: (1) structured attributes of the *metadata* of the video, (2) features extracted from raw *audio*, and (3) features extracted from raw *images* (thumbnails). Based on the preprocessing described in Section 3, we transform non-textual elements of a video (i.e., audio and images) into a text representation. Other textual elements such as the title of the video and the tags are kept as text. These textual representations are then transformed into a fixed-size input space vector of categorical features. This is done by tokenizing the input text to obtain a nominal discrete representation of the words described on it. Thus, feature vectors will have a limited number of possible values given by the bag of words representing the corpus in the training set. When extracting features from the text, we count the frequency that a word appears in the text.

Since in large text corpus certain words–e.g., articles–can appear rather frequently without carrying meaningful information, we transform occurrences into a score based on two relative frequencies known as *term-frequency* and *inverse document-frequency* (*TF-IDF*s). Intuitively, the term frequency represents how "popular" a word is in a text (in the feature vector), and the inverse document-frequency represents how "popular" a word appears, provided that it does not appear very frequently in other in the corpus (the feature space). More formally, we compute as $idf(t) = log\frac{1+n_s}{1+df(s,t)} + 1$, where n_s is the total number of samples and $df(s,t)$ is the number of samples containing term t.

As for the thumbnails, after extracting the most representative descriptions per image, we remove the least informative elements and only retain entities (nouns), actions (verbs), and modifiers (adverbs and adjectives). Each element in the caption is processed to a common base to reduce inflectional forms and derived forms (known as stemming). Further, we abstract the descriptions obtained from the images using *topic-modeling* as described earlier.

In our implementation, we extract features from only one image of the video (i.e., the thumbnail). Again, this is mainly because the thumbnails are typically purposely selected by the user and encapsulate semantically relevant context. However, we emphasize that our architecture could support the extraction of features from *every* frame in the video.

4.3 Prediction Models

We use three independent classifiers to estimate the likelihood of a video being targeted by hate attacks. These are built to operate independently, possibly when a new video is uploaded. Each classifier is designed to model traits from different aspects of the video as described above. Available decisions are later combined to provide one unified output.

We use three different classifiers, in an ensemble, because features obtained from different parts of a video are inherently incomplete, as some fields are optional and others might not be meaningful for certain video. For instance, a music video might not report a lengthy transcript, or a thumbnail might not contain distinguishable context. Since any reliable decision system should be able to deal with incomplete data, ensemble methods are well-suited to this setting. Moreover, ensembles often perform better than single classifiers overall [16].

4.3.1 Metadata and thumbnail classifiers

We build a prediction model such that $P_i(X = raid)$ based on the features extracted from the metadata (P_M) and from the image thumbnails (P_I). The architecture of these two predictors is flexible and accepts a range of classifiers. Our current implementation supports Random Forests (RFs), Extra Randomized Trees (XTREEs), and Support Vector Machines (SVM), both radial and linear. For our experiments, we select RF as the base classifier for P_T and SVM with linear kernel for P_M. Both SVM and RF have been successfully applied to different aspects of security in the past (e.g., fraud detection [7]) and have been shown to outperform other classifiers (when compared to 180 classifiers used for real-world problems [22]).

4.3.2 Audio-transcript classifier

Before feeding the transcripts to the classifier, we remove any words that have a transcription confidence $p_{trans} < 0.5$ as these are likely incorrectly transcribed and unrelated to the video context (including them will only confuse the classifier). Note that this only removes 9.2% of transcribed words. Additionally, the transcripts contain a lot of repeated terms that are mostly exclamations such as "uh uh," or "hm hm," Finally, the transcripts contain tags for non-verbal communication such as noise, laughter, etc., which we leave in the text as they do carry predictive power.

There are several choices in terms of selecting a classifier for long sequences of text. We experiment with transitional TF-IDF based approaches, Convolutional Networks, and Recurrent Neural Networks (RNN), ultimately opting for the latter since it yields the best performance and is quite effective at understanding sequences of words and interpreting the overall context. We also use an attention mechanism [4] as this can help an RNN "focus" on sequences of words that might indicate potentially raided videos. Before feeding any text to the network, we also need to transform each transcript to a sequence of words. Because neural networks process data in mini-batches, all transcripts within a mini-batch must have the same length (number of words). Transcripts with more words than the sequence length will be trimmed whereas samples with less words are left-padded with zeros (the model will learn zeros carry no information). Ideally, we want to setup a sequence length that is large enough to contain the text from all samples in a mini-batch but not too long to waste resources (feeding zeros in the network). We thus take the 95th percentile of length (with respect to the number of words) in the input corpus as the optimal sequence length (i.e., 5% of samples will be truncated). This results in a sequence length of 2,500 words.

The first layer of the network performs a word embedding, mapping each word to a high-dimensional vector. We do this because word embedding is effective for text classification tasks, especially when having relatively few training samples. We use pre-trained word embeddings from GloVe, which was constructed on more than 2 billion tweets that map each word into a 200-dimension vector. If a word is not found in the GloVe dataset, we initialize a vector of random weights, which the word embedding layer eventually learns from the input data. Next, after experimenting with several choices for the RNN architecture we use a layer of 256 GRU units. To reduce over-fitting we use a recurrent dropout with $p = 0.5$ as it empirically provided the best results across our datasets. On top of the recurrent layer, we add an attention layer as we are working with large sequence lengths (2,500 words). The network at this stage outputs one activation at the end of the whole sequence (the whole transcript).

We then add a fully-connected (dense) layer to the recurrent part to mix the recurrent layer outputs and bring the dimensionality down to 128 units. The output layer is another dense layer with one neuron per class. We use softmax as the activation function to normalize output values between 0 and 1.

For training, we use mini-batches of 32 transcripts (i.e., the input is of shape 32x2,500). We use *categorical cross-entropy* as a loss function and *Adam* as the optimization function. We allow a maximum of 100 epochs and use a separate validation set to perform *early stopping*: training is interrupted if the validation loss does not drop over 10 consecutive epochs, at which point the weights of the best epoch are restored. Finally, we note that our implementation uses Keras [36] with Theano [64] as a back-end.

4.3.3 Ensemble classifier

The objective of the ensemble method is to aggregate the predictions of the different base estimators. Each classifier individually models the likelihood that a video will be be targeted by hate attacks based on the set of features. The idea is that the decisions are then combined together so that the ensemble is able to make a more informed prediction. This not only allows for more robust predictions (in terms of confidence), but can also result in a more accurate prediction. We design our ensemble method to take a weighted vote of the available predictions. To compute the best-performing set of weights, we estimate a function f that takes as input each of the individual probabilities and outputs the aggregated prediction. During training this function learns from an independent testing set, and will be used during testing to weight each prediction model P_i. Formally,

$$f(P_M, P_I, P_T) = \{\texttt{raid}, \texttt{non-raid}\}.$$

For the decision function f of our *weighted-vote* algorithm (see Section 4.1), we use a distribution function that models how an expected probability in the testing set is affected by individual decisions P_i in a multiple regression. This function is approximated in the following form:

$$f = \sum_{j=1}^{n} w_j p_{ij},\ i = 1, \cdots, p$$

where p is the number of individual estimators and n is the number of observations in the validation set. This can be interpreted as the sum of the wights w times the probability score $p_i \in P_i$ given by the individual classifiers in the weighted voting system.

In our implementations, we use different underlying classification algorithms for estimating f. However, in the next section, we only present the results for each of the individual classifiers and two ensemble methods, namely, *weighted-vote*, and *average-prediction*. For the former, weights are fit using XTREE [23]. For the latter, we also test different settings, in particular, we try settings where one of the classifier is given a fixed weight of $w = 0$ and we average the others. The main reason is to understand how each of the individual classifiers influence each other, though we only report results for the best setting.

5 Evaluation

In this section, we present the setup and the results of our experimental evaluation.

5.1 Experimental Setup

Our main objective is to show that we can distinguish between raided and non-raided videos. However, there are several subtasks we also want to evaluate, aiming to better characterize the problem and understand how our classifiers perform.

Experiments. We start by trying to distinguish between random YouTube videos and those that are linked from /pol/. Next, we distinguish between the videos that are raided and those that are not (whether posted on /pol/ or not). Finally, we predict whether videos posted on 4chan will be raided.

More specifically, in EXPERIMENT 1, we set out to measure whether our classifiers are able to distinguish between videos linked from /pol/ and a random video uploaded to YouTube, aiming to gather insight into the ability to discriminate between videos *potentially* raided vs. those that are not at risk at all. Then, EXPERIMENT 2 evaluates whether or not the classifier can distinguish between any non-raided video (i.e., regardless of whether it is a random YouTube video or one posted on 4chan) and videos that will be raided. Finally, in EXPERIMENT 3, we focus on videos posted on 4chan, and determine which are going to be raided and which are not; this ensures that we can not only predict whether a video was posted on 4chan, but whether or not the video will be raided.

Train, Test, and Validate Splits. We split our datasets into three chunks: two for training and tuning parameters of the ensemble (training and testing) and one for validation, and report performance metrics on the latter. As we are dealing with highly unbalanced classes (there are multiple orders of magnitude more videos posted to YouTube than those posted to 4chan, let alone those that are raided), we balance the training and testing sets to model both classes properly, but leave the validation set unbalanced. Leaving the training split unbalanced would make it difficult for our models to properly learn the differences between the different classes. The validation set, however, remains unbalanced to more realistically model a real-world scenario.

The total number of videos in each split is proportionally sampled depending on the less populated class, assigning splits of 60%, 20%, and 20% to the training, testing, and validation sets. The more populated class uses the same amount of samples for training and test, while it will have all the remaining samples in the validation set. This procedure is repeated 10 times and the results are an average of the 10 different rounds. Table 3 summarizes all settings in our experiments, along with the number of samples used.

Evaluation Metrics. We evaluate our system using accuracy, precision, recall, and F1-measure. Precision measures the performance of our algorithm only for the values of the class of interest, while recall measures the proportion of positives that are correctly identified as such. The F1-measure is the harmonic mean of precision and recall. Finally, accuracy quantifies the proportion of correct predictions made in both classes.

Overall, these metrics are a good summary of the performance of an classifier in terms of True Negatives (TN), False Negatives (FN), False Positives (FP), and True Positives (TP); however, they are not ideal for comparing results across differ-

ID	Description	Training	Test	Validation
Exp. 1	Random YouTube vs. all on 4chan	731+731	243+243	13,470+244
Exp. 2	All non-raided vs. raided on 4chan	258+258	85+85	14,890+85
Exp. 3	Non-raided on 4chan vs. raided on 4chan	258+258	85+85	446+85

Table 3: Number of samples used in our experiments. The sets are balanced as there is the same amount of samples per each class (*class 1 samples+class 2 samples*) in training and test, while they are unbalanced in the validation set.

ent experiments. Therefore, we will also plot the Area Under the Curve (AUC), which reports the TP-rate (*recall*) against the FP-rate (*1 - recall*).

5.2 Experimental Results

We now report the results of our experimental evaluations, as per the settings introduced above. To ease presentation, we only report metrics for the individual classifiers as well as two ensemble methods: *weighted-vote* and *average-prediction*. We do not report results for other ensemble classifiers (*simple-voting* and the other underlying algorithms for estimating the weights), since they under-perform in our experiments.

For *weighted-vote*, weights are fit using XTREE [23], as described in Section 4.3.3. Also note that, for *average-prediction*, we find that the thumbnails classifier tends to disagree with the metadata and the transcripts classifiers combined. Therefore, in this mode, we fix a weight of $w = 0$ for the thumbnails classifier (i.e., $w_{thumbnail} = 0$).

Experiment 1. As mentioned, in this experiment we study whether we can predict that a video is linked from 4chan. Results are reported in Table 4. Overall, we find that we can correctly identify 92% of the videos (see *average-prediction* ensemble), and maintain high recall. Since we are dealing with a rather unbalanced validation set (in favor of the negative class), it is not surprising that precision drops to values close to 0, even though we have high accuracy.

Looking at the results obtained by the individual classifiers, we note that metadata has the highest accuracy (0.91), although audio-transcript scores highly as well (0.81), with the weighted-vote ensemble classifier matching the best recall from metadata (0.91). The best AUC value is the same between the metadata classifier and the weighted-vote ensemble (0.96).

In Figure 5a, we also plot the ROC curve for all five classifiers. The individual AUC scores are 0.79, 0.96, 0.62 for the transcripts, metadata, and thumbnails, respectively, while the two ensembles (*weighted-vote* and *average-prediction*) score, 0.96 and 0.95, respectively. The *weighted-vote* ensemble has the highest AUC throughout most of the x-axis, although, the ROC curve essentially overlaps with that of the the metadata classifier. The two ensemble have different strengths: the *weighted-vote* ensemble has the highest recall and AUC values, but the *average-prediction* (with $w_{thumbnail} = 0$) has the highest accuracy, precision, and F1-measure.

Experiment 2. In Figure 5b, we report the AUC when classifying raided and non-raided videos—regardless of whether the latter are random YouTube videos or non-raided ones posted on 4chan. We find that the *average-prediction* ensemble classifier correctly labels 90% of the videos—as shown in Table 4). Unlike EXPERIMENT 1, among the individual classifiers, the best performance is achieved by the audio-transcript classifier, except for recall, where the metadata classifier performs best. This setting also yields high recall (0.88) when combining all classifiers into the *weighted-vote* ensemble. As in EXPERIMENT 1 the *weighted-vote* ensemble presents the highest recall and AUC, but the *average-prediction* has higher accuracy, precision, and F1-measure. Figure 5b shows a similar situation as in the previous experiment: the ROC curve for the metadata classifier is really close to or overlapping with the ones for the two ensemble. AUC equals to 0.61 for thumbnails, 0.79 for transcripts, and 0.94 for metadata. Whereas, the *weighted-vote* ensemble achieves 0.94 AUC as the metadata individual classifier, and *average-prediction* 0.92.

Experiment 3. Finally, we evaluate how well our models discriminate between raided videos posted to 4chan and non-raided videos also posted to 4chan. Our results confirm that this is indeed the most challenging task. Intuitively, these videos are much more similar to each other than those found randomly on YouTube. This is because /pol/ is interested in a particular type of content in general, regardless of whether or not the video ends up raided. Nonetheless, as shown in Table 4, around 75% of the videos are correctly classified by the best performing classifier, i.e., the *average-prediction* ensemble.

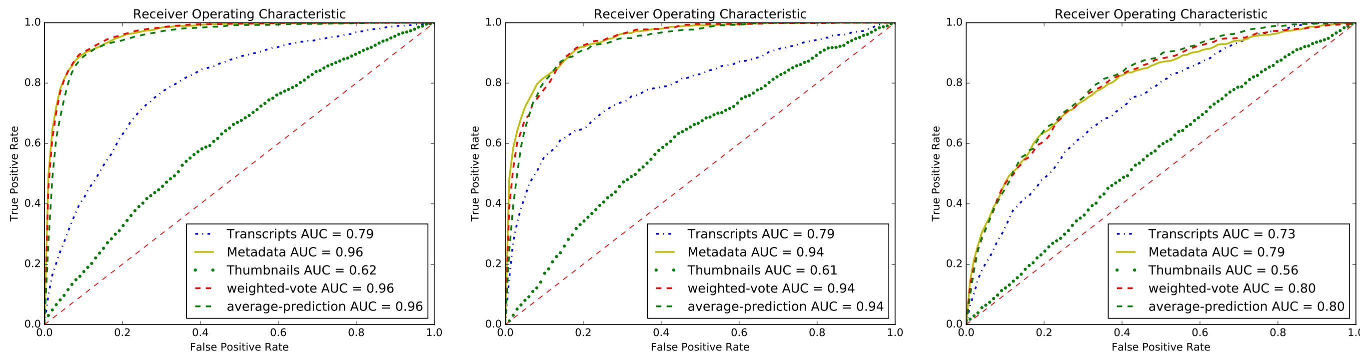
This setting shows a clear case for the ensemble classification yielding appreciably better performance. Overall, the individual classifiers, i.e., transcripts, metadata, and thumbnails reach AUCs of 0.73, 0.79, and 0.56, respectively, whereas, both the ensemble classifiers reach 0.80. Nevertheless, the ROC curve in Figure 5c shows how the *weighted-vote* ensemble is sometimes penalized by the weakest performing classifier (i.e., thumbnails classifier). This is apparent by comparing the differences between *weighted-vote* and *average-prediction* (recall that $w_{thumbnails} = 0$ in the latter).

5.3 Choosing an Ensemble

The goal of the our system is to make the best final "decision" possible given the choices made by the individual classifiers. In absolute terms, the *weighted-vote* (with XTREE as baseline estimator) yields the best performance in all three experiments in terms of recall (and overall AUC). In particular, it outperforms the *average-prediction* ensemble in two of the tasks: modeling videos from /pol/ (EXPERIMENT 1), and detecting raided videos in general (EXPERIMENT 2). When restricting our attention to the detection of raids of videos posted on 4chan (EXPERIMENT 3), both ensemble methods are comparable in terms of recall. However, when looking at precision, we find that *average-prediction* outperforms *weighted-vote*. The trade-off between having good precision and good recall will have an important impact on the amount of vetting work required by providers deploying our system, as we dis-

	Experiment 1					Experiment 2					Experiment 3				
Classifier	**ACC**	**PRE**	**REC**	**F1**	**AUC**	**ACC**	**PRE**	**REC**	**F1**	**AUC**	**ACC**	**PRE**	**REC**	**F1**	**AUC**
transcripts	0.81	0.05	0.60	0.10	0.79	0.89	0.03	0.56	0.06	0.79	0.71	0.32	0.58	0.40	0.73
metadata	0.91	0.13	0.89	0.23	**0.96**	0.87	0.03	0.85	0.06	**0.94**	0.73	0.32	**0.71**	0.44	0.79
thumbnails	0.55	0.02	0.64	0.05	0.62	0.52	0.01	0.66	0.02	0.61	0.53	0.18	0.55	0.27	0.56
weighted-vote ensemble	0.89	0.12	**0.91**	0.21	**0.96**	0.85	0.03	**0.88**	0.05	**0.94**	0.74	0.34	0.69	0.45	**0.80**
average-prediction ensemble	**0.92**	**0.15**	0.85	**0.26**	0.95	**0.90**	**0.04**	0.82	**0.07**	0.92	**0.75**	**0.35**	0.69	**0.46**	**0.80**

Table 4: Results for Experiment 1 (videos posted on 4chan and random YouTube samples), Experiment 2 (raided videos posted on 4chan and all non raided videos), and for Experiment 3 (non-raided videos posted on 4chan and raided videos posted on 4chan). ACC stands for accuracy, PRE for precision, and REC for recall. The ensemble classifiers have different inputs: the weighted-vote classifier receives inputs from all three the individual ones, while the average-prediction receives the inputs only from the metadata and the transcript classifier.



(a) Experiment 1 **(b)** Experiment 2 **(c)** Experiment 3

Figure 5: ROC curves for each experiment. AUC values for Thumbnails, Transcripts, Metadata, and Ensemble classifiers, XTREE and Average probabilities.

cuss in more detail in Section 6.

In the following, we provide an explanation as to why the two ensemble classifiers report similar results in some experiments and different ones in others. When using a base estimator to fit the best weights for the individual classifiers, we observe a bias towards the decisions made by the metadata classifier. This is expected, as this classifier is the one that performs best among the individual classifiers (and substantially so in both Experiment 1 and Experiment 2). On the contrary, the thumbnails classifier performs worst, except for recall in Experiment 2. As for the precision, the thumbnails classifier always perform the worst.

We highlight that our data includes videos with partially-available features. When this happens, the ensemble classifier is forced to make a decision based on the other inputs. This is precisely the case for the thumbnails, which are not always available. This is why we evaluated the *average-prediction* ensemble method forcing a weight $w_{thumbnails} = 0$. In this setting, the *weighted-vote* method with XTREE provided similar results, since XTREE initially assigned a low weight (although not exactly 0) to the thumbnails.

Overall, with the *average-prediction* method, accuracy is always better than for both the metadata and the XTREE ensemble classifiers. This also applies for precision and F1-measure. This means that this configuration reduces the number of false positives and, as a consequence, is slightly more accurate. In other words, this highlights how, when the individual classifiers have similar performance, the ensemble is better than the best options among the single classifiers. In fact, the metadata and transcripts classifiers have different performances, but the thumbnails classifier differs the most.

6 Discussion

The experiments presented above show that we can model the characteristics of YouTube videos which are likely to be raided by users of third-party hateful communities. In other words, the results make an encouraging case that automated machine learning techniques can be used to prevent or mitigate the effects of aggression on video sharing platforms. What still needs to be ironed out is how our techniques could be integrated and deployed by mainstream providers like YouTube. Although a definitive path to adoption is out of scope for this paper, we discus this aspect a bit next.

We start by noting that our system is really geared to identify videos that are *at risk* of being raided, thus, YouTube could use it as a *filter*, i.e., flagging videos that are risky and/or might require particular attention. The "risk" can be derived from the probability output of the ensemble, or any of the individual classifiers, and adjusted to minimize the number of missed detections (videos that will be raided but that the system has not flagged), i.e., maintaining high recall. Tuning the model to balance the overall number of flagged videos as well ensuring that they are indeed at high risk can help reduce the impact of false positives. Given that our datasets are extremely unbalanced, high precision, on the other hand, is not a top priority. As mentioned above, the system would flag videos likely to be raided, thus, helping to tackle the problem of aggression by reducing the videos that need to be monitored. While this would certainly reduce the amount of human labor involved in dealing with raids, it could also be used to focus costlier detection systems on high risk videos.

Our evaluation also showed that we can distinguish videos that are likely to be raided from regular YouTube videos with high accuracy. This means that YouTube could deploy our system at *upload time*, determining the likelihood of a video to be raided at some point in the future. The platform could then adopt mitigation strategies for videos that prove risky, for example by manually moderating the comments to such videos. This is a practice already employed by YouTube [52], however, at the moment, the efficacy of this moderation has been questioned [21] due to the sheer volume of content, in addition to YouTube's focus on removing *inappropriate* videos instead of protecting users against raids.

We estimate, however, that only 16% of videos would require any action—an estimation based on EXPERIMENT 2. While this might appear to be a very large number, it is still less than having to monitor all videos. Moreover, additional automated tools could be deployed to check whether a raid is actually occurring before being passed along for human review. Furthermore, YouTube has recently pledged to hire 10,000 new workers to monitor content [21], but deploying our system could reduce the need for this many humans, or at worst, allow them to focus on higher impact content.

Then, EXPERIMENT 3 demonstrates that, when provided with videos linked from fringe communities such as /pol/, our system can successfully identify those videos that are likely to be raided with reasonable accuracy. This is a much more difficult task, and thus accuracy is lower, since videos are very often being posted to /pol/ without the actual intent of having raiders show up in the first place. Furthermore, the number of videos posted on /pol/ is much smaller than those uploaded to YouTube as a whole—for instance, the /pol/ dataset from [30] contains links to 93K YouTube videos posted over a period of 3 months. Among these links, we only selected those that had crystal clear evidence of raids by restricting the thresholds of HCPS and time lag (see Section 2.2), also discarding videos which could have been raided from the non-raided group. This choice is extremely conservative (yielding 428 videos), aiming to have a reliable ground truth on which to evaluate the system. Although we could have relaxed our thresholds a bit and obtained higher accuracy, the applicability to real-world use cases would likely have been affected, as our ground truth dataset would have included videos that had controversial or combative statements, but were not actually raided.

Also note that /pol/, though a very good example of a tight-knit community used to coordinate and disrupt other social groups, is not the only community responsible for performing raids against YouTube videos. Other Web communities, e.g,. Reddit [54] or Kiwi Farms [20] also regularly take part in raiding activity. The same techniques presented here, however, can be used to detect raids from other communities.

Finally, it might perhaps be tempting to dismiss the relatively low occurrence of raids, vis-à-vis the number of YouTube videos posted every day, as being a niche problem. On the contrary, harassment and bullying on YouTube are widely recognized as a serious issue by authorities on the matter [47], and news reports are filled with ghastly stories [50] and advice on how to deal with hateful and harassing YouTube comments in particular [15, 26]. Although we are not aware of any suicide cases directly linked to YouTube raids, victims have indeed been active on YouTube [48, 29], and thus raids pose very serious safety risks. Overall, even if the majority of content on YouTube (and other social media platforms) tends to be "safe," we should not discard the outsized effects that this negative behavior has. From a purely pragmatic point of view, advertisers providing the primary income stream for sites like YouTube have been re-thinking their reliance on social media in light of the recent surge in anti-social behavior [58]. From a societal point of view, raiding behavior is a pressing concern; it is a direct threat to free speech and civil discourse, and causes emotional distress that can lead to dire consequences. The efforts of the research community have enabled the long tail of the Web to succeed, building technologies that democratized information and shrunk the world. Thus, while raids on YouTube videos do occur in the long tail, we argue that dismissing them as being too rare is an abdication of our social responsibility.

7 Related Work

YouTube is used every day by millions of individuals to share various kinds of videos, e.g., music, lectures, gaming, video blogs, etc. [51]. Organized groups are also active on the platform; some use it to reach and grow their network of support and organize activities, while others to propel radicalization and initiatives against other groups. Previous work has looked at the use of YouTube by LGBT users for self-disclosure [27], for anti- or pro-anorexia [49], fat stigmatization [32], sharing violent content [67], far-right propaganda [18], as well as Jihad and self-radicalization [13]. These topics often attract considerable attention and typically lead to unwanted behavior studied, e.g., in the context of swearing, abusive, hateful, or flaming comments, as well as activity against the video poster and its supporters [43, 33].

Prior work has studied controversial YouTube videos aiming to understand what types of comments and user reaction certain categories of videos attract. For instance, [3] measure civic behavioral intention upon exposure to highly or lowly arousing videos showing cyberbullying activity, while [40] study user-posted "ranting" videos, which, although appearing to be aggressive, actually cause subtle differences in how they engage other users. Researchers have also focused on bullying and hate speech on, e.g., Twitter [9, 11] or Yahoo News and Finance [46]. By contrast, we focus on hateful activity against YouTube videos, and in particular on studying the types of videos that are more likely to be targeted by attacks.

Recent studies also analyze YouTube videos' comments to detect hate speech, bullying, and aggression via swearing in political videos. [38] and [39] investigate whether aggressive behavior (in online comments) can be contagious, observing mimicry of verbal aggression via swearing comments against Donald Trump's campaign channel. Interestingly, this aggressive emotional state can lead to contagious effects through textual mimicry. In this paper, we build on previous work on characterizing raiding behavior on YouTube, presenting a data-driven approach to identify *which* videos are likely to be the target of a raid.

Another line of work has looked at offensive or harmful YouTube videos and how to automatically detect them. This is an orthogonal problem to ours, as we look at videos posted with a legitimate purpose, that are later victim of coordinated attacks. Sureka et al. [61] use social network analysis to identify extremist videos on YouTube, while [2] detects violent and abusive videos, by mining the video's metadata such as linguistic features in the title and description, popularity of video, duration and category. Finally, Agarwal and Sureka [1] search for malicious and hateful videos using a topical crawler, best-first search, and shark-search for navigating nodes and links on YouTube.

Moreover, Marathe and Shirsat [42] study detection techniques used for other problems, e.g., spam detection, and assess whether they could be applied to raids. Also, Dadvar et al. [14] use machine learning to detect YouTube users exhibiting cyberbullying behavior. Whereas, rather than focusing on single offending users, we look at videos that are likely to receive hate and raids and their attributes from various users. Finally, Hine et al. [30], as already discussed, show that underground forums such as 4chan organize raids to platforms like Twitter, Google, and YouTube.

Overall, to the best of our knowledge, we are the first to study video properties, such as their transcripts, metadata, and thumbnails, to shed light on the characteristics of the videos raided by the users of such platforms, using advanced machine and deep learning techniques to perform detection of videos targeted by raids.

8 Conclusion

This paper presented a supervised learning based approach to automatically determine whether a YouTube video is likely to be "raided," i.e., receive a sudden spike in hateful comments as a result of an orchestrated effort coordinated from another platform. Our experimental results showed that even single-input classifiers that use metadata, thumbnails, or audio transcripts can be effective, and that an ensemble of classifiers can reach high detection performance, thus providing a deployable early-warning system.

Overall, our work represents an important first step toward providing video platforms like YouTube with proactive systems geared to detect and mitigate coordinated hate attacks. We discussed potential deployment strategies that could be taken by YouTube (or other providers), i.e., running our tool on every video at upload time and/or monitoring fringe communities such as 4chan to screen videos that are linked to on those platforms.

Note that the classifiers presented in this paper are not meant to provide a mechanism for *censoring* content or users, nor to identify users possibly involved in raids. Rather, we aim to identify content that is at risk of attack; once identified, *proactive* solutions to protect against raiders can be taken by the service providers. While the specifics are beyond the purpose of this paper, we believe that there are actions that can be taken that protect freedom of expression while also preserving civil discourse. For example, temporarily disabling or rate limiting comments, requiring new comments to be approved before going live, or simply notifying the poster that a raid might be coming could serve to balance protection vs. expression.

As part of future work, we plan to use deep-learning methods to fuse audio, video, and metadata into a single classifier. We also plan to look into raids from other communities, such as Reddit, Gab.ai, and Kiwi Farms.

Acknowledgments. This project has received funding from the European Union's Horizon 2020 Research and Innovation program under the Marie Skłodowska-Curie ENCASE project (GA No. 691025) and from the EPSRC (grant number EP/N008448/1). Enrico Mariconti was supported by the EPSRC under grant 1490017.

References

[1] S. Agarwal and A. Sureka. A focused crawler for mining hate and extremism promoting videos on youtube. In *ACM Hypertext*, 2014.

[2] N. Aggarwal, S. Agrawal, and A. Sureka. Mining YouTube metadata for detecting privacy invading harassment and misdemeanor videos. In *PST*, 2014.

[3] S. Alhabash, J. hwan Baek, C. Cunningham, and A. Hagerstrom. To comment or not to comment?: How virality, arousal level, and commenting behavior on YouTube videos affect civic behavioral intentions. *Computers in Human Behavior*, 51:520–531, 2015.

[4] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv:1409.0473*, 2014.

[5] F. Benevenuto, G. Magno, T. Rodrigues, and V. Almeida. Detecting spammers on twitter. In *CEAS*, 2010.

[6] F. Benevenuto, T. Rodrigues, V. Almeida, J. Almeida, and M. Gonçalves. Detecting spammers and content promoters in online video social networks. In *SIGIR*, 2009.

[7] S. Bhattacharyya, S. Jha, K. Tharakunnel, and J. C. Westland. Data mining for credit card fraud: A comparative study. *Decision Support Systems*, 50(3):602–613, 2011.

[8] A. Breland. The Hill – YouTube cracking down on hate speech. http://bit.ly/2wjoI47, 2017.

[9] P. Burnap and M. L. Williams. Us and them: identifying cyber hate on twitter across multiple protected characteristics. *EPJ Data Science*, 5, 2016.

[10] Q. Cao, X. Yang, J. Yu, and C. Palow. Uncovering large groups of active malicious accounts in online social networks. In *CCS*, 2014.

[11] D. Chatzakou, N. Kourtellis, J. Blackburn, E. De Cristofaro, G. Stringhini, and A. Vakali. Hate is not Binary: Studying Abu-

sive Behavior of #GamerGate on Twitter. In *ACM Hypertext*, 2017.

[12] S. Chess and A. Shaw. A conspiracy of fishes, or, how we learned to stop worrying about #GamerGate and embrace hegemonic masculinity. *Journal of Broadcasting & Electronic Media*, 2015.

[13] M. Conway and L. McInerney. *Jihadi Video and Auto-radicalisation: Evidence from an Exploratory YouTube Study*, pages 108–118. Springer Berlin Heidelberg, 2008.

[14] M. Dadvar, R. Trieschnigg, and F. de Jong. Experts and machines against bullies: A hybrid approach to detect cyberbullies. In *Canadian AI*, 2014.

[15] E. Denton. YouTuber Jaclyn Hill Reveals She's "Scared" of Her Channel Because the Comments Are So Mean. https://bit.ly/2HXSTa4, 2015.

[16] T. G. Dietterich. Ensemble Methods in Machine Learning. In *First International Workshop on Multiple Classifier Systems*, 2000.

[17] D. B. Eichenberger. Speech activity detection: Application-specific tuning and context-based neural approaches. Bachelor thesis, Universitat Politcnica de Catalunya, July 2016.

[18] M. Ekman. The dark side of online activism: Swedish right-wing extremist video activism on youtube. *MedieKultur: Journal of media and communication research*, 30(56):21, 2014.

[19] M. ElSherief, S. Nilizadeh, D. Nguyen, G. Vigna, and E. Belding. Peer to Peer Hate: Hate Speech Instigators and Their Targets. In *ICWSM*, 2018.

[20] K. Farms. Kiwi farms. https://kiwifarms.net/, 2018.

[21] B. Feldman. Can 10,000 Moderators Save YouTube? http://nymag.com/selectall/2017/12/can-10-000-moderators-save-youtube.html, December 2017.

[22] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim. Do we need hundreds of classifiers to solve real world classification problems? *JMLR*, 15(1):3133–3181, 2014.

[23] P. Geurts, D. Ernst, and L. Wehenkel. Extremely randomized trees. *Machine learning*, 63(1):3–42, 2006.

[24] J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone Speech Corpus for Research and Development. In *ICASSP*, 1992.

[25] V. Goel, S. Kumar, and W. Byrne. Segmental Minimum Bayes-Risk Decoding for Automatic Speech Recognition. *IEEE Transactions on Speech and Audio Processing*, 12(3):234–249, 2004.

[26] P. Gomez. Youtube and instagram stars explain how to protect your kids from online bullying. http://people.com/social-media-stars/protect-kids-online-bullying-youtube-instagram/, 2017.

[27] M. Green, A. Bobrowicz, and C. S. Ang. The lesbian, gay, bisexual and transgender community online: discussions of bullying and self-disclosure in YouTube videos. *Behaviour & Information Technology*, pages 1–9, 2015.

[28] D. W. Grigg. Cyber-aggression: Definition and concept of cyberbullying. *Australian Journal of Guidance and Counselling*, 20(2), 2010.

[29] G. Herbet. Hannah Stone suicide: Parents say YouTube rival's success devastated CNY singer . https://www.syracuse.com/expo/erry-2018/05/f63834778d4104/hannah_stone_suicide_danielle.html, May 2018.

[30] 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.

[31] B.-J. P. Hsu and J. R. Glass. Iterative language model estimation: efficient data structure & algorithms. In *Interspeech*, 2008.

[32] M. Hussin, S. Frazier, and J. K. Thompson. Fat stigmatization on youtube: A content analysis. *Body Image*, 8(1):90 – 92, 2011.

[33] L. Jnson. Flaming motivation in YouTube users as a function of the traits Disinhibition seeking, Assertiveness and Anxiety? Technical report, University of Twente, 2013.

[34] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015.

[35] I. Kayes, N. Kourtellis, D. Quercia, A. Iamnitchi, and F. Bonchi. The Social World of Content Abusers in Community Question Answering. In *WWW*, 2015.

[36] Keras. Keras: The python deep learning library. https://keras.io/, 2018.

[37] S. Kumar, W. L. Hamilton, J. Leskovec, and D. Jurafsky. Community interaction and conflict on the web. In *The Web Conference (WWW)*, 2018.

[38] K. Kwon and A. Gruzd. Is Aggression Contagious Online? A Case of Swearing on Donald Trump's Campaign Videos on YouTube. In *Hawaii International Conference on System Sciences*, 2017.

[39] K. H. Kwon and A. Gruzd. Is offensive commenting contagious online? examining public vs interpersonal swearing in response to donald trump's youtube campaign videos. *Internet Research*, 2017.

[40] P. G. Lange. Commenting on YouTube rants: Perceptions of inappropriateness or civic engagement? *Journal of Pragmatics*, 2014.

[41] J. Luque, C. Segura, A. Snchez, M. Umbert, and L. A. Galindo. The role of linguistic and prosodic cues on the prediction of self-reported satisfaction in contact centre phone calls. In *Proc. Interspeech 2017*, pages 2346–2350, 2017.

[42] S. Marathe and K. P. Shirsat. Approaches for mining youtube videos metadata in cyber bullying detection. *International Journal of Engineering Research & Technology*, 4:680–684, 2015.

[43] P. J. Moor, A. Heuvelman, and R. Verleur. Flaming on youtube. *Computers in Human Behavior*, 26(6):1536–1546, November 2010.

[44] B. Moser. How YouTube Became the Worldwide Leader in White Supremacy. https://goo.gl/4bMjXn, 2017.

[45] S. Nilizadeh, F. Labrèche, A. Sedighian, A. Zand, J. Fernandez, C. Kruegel, G. Stringhini, and G. Vigna. Poised: Spotting twitter spam off the beaten paths. In *CCS*, 2017.

[46] C. Nobata, J. Tetreault, A. Thomas, Y. Mehdad, and Y. Chang. Abusive Language Detection in Online User Content. In *WWW*, 2016.

[47] Nobullying.com. Youtube Bullying. https://nobullying.com/youtube-bullying/, 2018.

[48] A. O'Connor. Suicide Draws Attention to Gay Bullying. https://well.blogs.nytimes.com/2011/09/21/suicide-of-gay-teenager-who-urged-hope/, September 2011.

[49] A. Oksanen, D. Garcia, A. Sirola, M. Nsi, M. Kaakinen,

T. Keipi, and P. Rsnen. Pro-Anorexia and Anti-Pro-Anorexia Videos on YouTube: Sentiment Analysis of User Responses. *Journal of Medical Internet Research*, 17, October 2015.

[50] M. Oppenheim. Jessi Slaughter on becoming a meme and falling victim to trolls after infamous YouTube video. https://ind.pn/2IjeRnB, March 2016.

[51] J. Y. Park, J. Jang, A. Jaimes, C.-W. Chung, and S.-H. Myaeng. Exploring the User-generated Content (UGC) Uploading Behavior on Youtube. In *WWW Companion*, 2014.

[52] S. Perez. Youtube promises to increase content moderation and other enforcement staff to 10k in 2018. https://goo.gl/2te7HV, 2018.

[53] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. The Kaldi Speech Recognition Toolkit. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[54] A. Romano. Reddit just banned one of its most toxic forums. But it wont touch The_Donald. https://www.vox.com/culture/2017/11/13/16624688/reddit-bans-incels-the-donald-controversy, November 2017.

[55] C. Rossow. Amplification hell: Revisiting network protocols for ddos abuse. In *Network and Distributed Systems Security Symposium (NDSS)*, 2014.

[56] R. Speer, J. Chin, and C. Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451, 2017.

[57] Statista. Share of internet users in the United States who have personally experienced online harassment as of January 2017. https://www.statista.com/statistics/333942/us-internet-online-harassment-severity/, 2017.

[58] N. Statt. YouTube is facing a full-scale advertising boycott over hate speech. https://www.theverge.com/2017/3/24/15053990/google-youtube-advertising-boycott-hate-speech, March 2017.

[59] G. Stringhini, C. Kruegel, and G. Vigna. Detecting spammers on social networks. In *ACSAC*, 2010.

[60] G. Stringhini, P. Mourlanne, G. Jacob, M. Egele, C. Kruegel, and G. Vigna. Evilcohort: detecting communities of malicious accounts on online services. In *USENIX Security Symposium*, 2015.

[61] A. Sureka, P. Kumaraguru, A. Goyal, and S. Chhabra. Mining YouTube to Discover Extremist Videos, Users and Hidden Communities. In *AIRS*, pages 13–24, 2010.

[62] The Guardian. Twitter CEO: We suck at dealing with trolls and abuse. https://www.theguardian.com/technology/2015/feb/05/twitter-ceo-we-suck-dealing-with-trolls-abuse, 2015.

[63] The Guardian. Are Facebook and Twitter doing enough to protect users? https://www.theguardian.com/media-network/2016/aug/26/social-media-sites-protect-users-twitter-facebook, 2016.

[64] Theano. http://deeplearning.net/software/theano/, 2018.

[65] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2016.

[66] B. Viswanath, M. A. Bashir, M. Crovella, S. Guha, K. P. Gummadi, B. Krishnamurthy, and A. Mislove. Towards detecting anomalous user behavior in online social networks. In *USENIX Security Symposium*, 2014.

[67] A. Weaver, A. Zelenkauskaite, and L. Samson. The (Non)Violent World of Youtube: Content Trends in Web Video. *Journal of Communication*, 62(6), 2012.

[68] Wikipedia. Triple parentheses. https://en.wikipedia.org/wiki/Triple_parentheses, 2018.

[69] YouTube. Lesson: Make effective thumbnails and titles. https://creatoracademy.youtube.com/page/lesson/thumbnails#yt-creators-strategies-5, 2018.

[70] YouTube. Youtube for the press. https://www.youtube.com/yt/about/press/, 2018.