VulnerCheck: A Content-Agnostic Detector for Online Hatred-Vulnerable Videos

Lanyu Shang, Daniel (Yue) Zhang, Michael Wang, Dong Wang
Department of Computer Science and Engineering
University of Notre Dame, Notre Dame, IN, USA
{*lshang, yzhang40, mwang6, dwang5*}*@nd.edu*

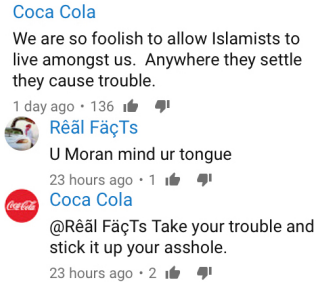
***Abstract*—With the increasing popularity of online video platforms (e.g., YouTube, Vimeo), the spread of hateful videos and the lack of rigorous hateful content control have become a critical issue. This paper focuses on the problem of identifying online *hatred-vulnerable* videos where the videos themselves do not contain any hateful content but unexpectedly trigger hateful comments from the audience. It is suboptimal to simply treat the hatred-vulnerable videos as hateful ones and remove them from the sharing platforms. This will discourage the uploaders of such videos from sharing valid and informative videos in the future. However, treating these hatred-vulnerable videos as hatred-free ones will provide undesirable opportunities for hateful users to spread their toxic comments and extreme ideology. In this paper, we develop VulnerCheck, an end-to-end supervised learning approach to effectively classify hatred-vulnerable videos from hateful and hatred-free ones by exploring the structure and semantics features of audience's comment networks. VulnerCheck is content-agnostic in the sense that it does not analyze the content of the video and is therefore robust against sophisticated content creators who craft hateful videos to bypass the current content censorship. We evaluate VulnerCheck on a real-world dataset collected from YouTube. Results demonstrate that our scheme is both effective and efficient in identifying hatred-vulnerable videos and significantly outperforms the state-of-the-art baselines.**

1. Introduction

In recent years, the spread of hateful online videos and the lack of rigorous hateful content control mechanisms have raised many concerns in our society [1], [2]. The online video sharing platforms have striven to combat hateful content by leveraging both automatic hate speech detection systems and voluntary reports from the audience [3]. While those solutions primarily focus on minimizing the number of hateful videos that go undetected by their systems [4], they are often known to suffer from high false positives (e.g., mistakenly identifying valid videos that do not violate any laws and regulations as hateful videos) [5]. In this paper, we focus on the problem of identifying *online hatred-vulnerable* videos where the content of the video is hate-free, but they unexpectedly trigger or attract hateful comments from users with a certain ideology, religion, or cultural background.

Figure 1 shows an example of a hatred-vulnerable video[1] and example comments from its audience. The video is a neutral news report in the aftermath of the suicide bombing event in Sri Lanka on April 21, 2019. However, we find a non-trivial amount of hateful and extreme comments against Islam from the audience. On the one hand, simply removing the hatred-vulnerable videos from online video sharing platforms (e.g., YouTube) is suboptimal: it will discourage uploaders (e.g., news channel in the above example) from publishing and sharing valid and informative videos on the platforms in the future. On the other hand, treating the hatred-vulnerable videos as normal ones will provide undesirable opportunities for hateful users to spread their toxic comments and extreme ideology. Therefore, it is critical to develop an effective mechanism to detect hatred-vulnerable videos and suppress the propagation of hateful comments through such videos.

(a) Video



(b) Comments

Figure 1: An example of Hatred-vulnerable Videos

The detection of hatred-vulnerable video is beneficial in several aspects. First, it can greatly reduce the propagation of hateful ideology on online platforms [6]. For example, the video sharing platforms could remind the audience to be mindful when watching the identified hatred-vulnerable videos (e.g., adding a vulnerable badge or banner to the video), or browsing/posting comments to these videos (e.g., demoting extreme comments with hateful speech, promoting comments countering the hateful ones). Second, the identification of hatred-vulnerable videos also provides new opportunities to efficiently spot "hidden evils" who generate and

[1] Video link: https://www.youtube.com/watch?v=Gi1xfFEMhSQ

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

spread hateful and seditious content through their comments on the hatred-vulnerable videos [7].

The detection of hatred-vulnerable video requires a careful analysis of the hatefulness of both the video content and its comments (as shown in Figure 2). Many solutions [1], [6], [8], [9], [10]have been developed to identify hatred and extremism information online. However, those solutions cannot be directly applied to solve our problem. First, it is challenging to distinguish the *hatred-vulnerable* videos from the *hateful* ones. This is mainly because the hateful content is an abstract concept that can be hardly identified by video content-based approaches that process the visual and auditory content of the video. For example, the image-based video classification approaches [8] often focus on object detection and action recognition (e.g., pornography, violent scenes) and are insufficient to capture the abstract idea of hatred-related content. Additionally, speech recognition approaches [11] usually convert speeches in the audio track to textual content to extract video content-related features (e.g., document embedding, n-grams). However, a significant amount of hateful videos are intentionally crafted by sophisticated creators [12] to circumvent the audio detection. Typical examples include videos that only have background music or contain multiple audio tracks [13].

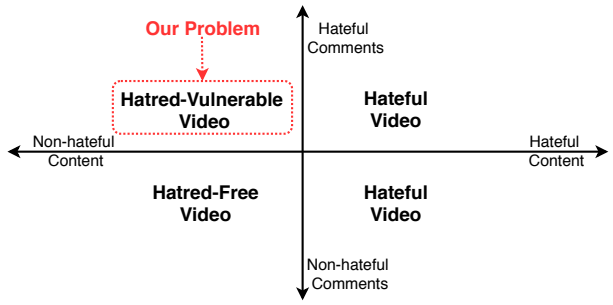


Figure 2: Video Categories

It is also a challenging problem to differentiate *hatred-vulnerable* videos from the *hatred-free* ones based on the hatefulness of video comments. Efforts have been made to detect and prohibit hate speech on online media [1], [6], [9]. While these approaches are effective in detecting hateful comments from individual users, they largely ignore the complex interactions between users who comment on the videos. For example, the top-level comment in Figure 1(b) may go undetected by the current detection schemes since the comment itself does not contain any hateful vocabulary. However, the idea that the comment tries to express indeed triggered and attracted hateful comments against Islam. In addition, it is a non-trivial task to distinguish between hate speech and impolite language in user comments. For example, offensive phrases (e.g., "h*e" and "bi*ch") are commonly found in rap lyrics and comments quoting such lyrics are often classified as hate speech by current solutions [14]. Therefore, the detection of hatred-vulnerable videos cannot be simply addressed by the video content-based or the hateful speech-based solutions.

In this paper, we develop VulnerCheck, an end-to-end supervised learning approach that can effectively identify the online hatred-vulnerable videos on video sharing platforms. VulnerCheck is a content-agnostic approach that does not directly analyze the video content. Instead, it leverages the intelligence and interactions from the audience of the video by exploring the structure and semantic features of their comment networks. In particular, VulnerCheck constructs a novel semantic-aware comment network to effectively capture the commenting behavior of users through both the topological and semantic features of the comment network. It also extracts relevant linguistic features of user comments and the metadata features of the video. The extracted features are integrated with a supervised classification framework that can accurately classify hatred-vulnerable videos from hateful and hatred-free ones. To the best of our knowledge, VulnerCheck is the first solution to address the online hatred-vulnerable video detection problem using a content-agnostic approach. We evaluate VulnerCheck with a real-world video dataset collected from YouTube. The results show that VulnerCheck significantly outperforms the state-of-the-art baseline methods in detecting the hatred-vulnerable videos.

2. Related Work

2.1. Hate Speech Detection

The spread of hate speech against disadvantaged groups has become a severe issue on online social media [15], [14], [16], [17], [18]. A significant amount of efforts have been made to address this problem. For example, Badjatiya *et al.* developed a deep learning based hate speech detection tool for tweets [19]. It focused on learning semantic word embeddings from a large set of annotated tweets through multiple deep learning architectures. Waseem *et al.* developed a racist-related hate speech detector based on the critical race theory and the n-gram features extracted from a large annotated Twitter dataset [9]. Davidson *et al.* further developed a machine learning-based approach to automatically identify the hate speech using a crowd-sourced hate speech lexicon collected from social media [14]. While the above schemes can effectively detect hate speech from online social media posts, they are insufficient to address the online hatred-vulnerable video detection problem because both hateful and hatred-vulnerable videos appear to be attractive to hate speech in the video comments.

2.2. Counterspeech Detection

Counterspeech, as a type of response to extremism or hate speech, has emerged as an alternative approach to combat the hateful content on online platforms [20], [21]. For example, Bartlett *et al.* studied the spread of hate speech and counterspeech on Fackbook and analyzed their characteristics (e.g., propagation pattern, user interaction) [20]. Schieb *et al.* evaluated the impact of counterspeech in prohibiting

hate speech with a focus on a set of influential factors (e.g., opinion, volatility, and user participation) [22]. In addition, Mathew *et al.* analyzed the hate speech and counterspeech behavior of Twitter users and extracted a group of comprehensive features (e.g., user profile properties and post lexical properties) to identify hateful Twitter accounts [23]. However, these solutions primarily focus on the difference between hate speech and counterspeech on the linguistic level, which is shown to be insufficient to identify online hatred-vulnerable videos. In contrast, our work jointly considers the hate speech, counterspeech, and their interactions in the video comments. In particular, VulnerCheck captures both semantic features and topological characteristics of the user comment networks.

2.3. Hateful Video Detection

The detection of hateful content in videos is also related to our problem. For example, Rafiq *et al.* developed a cyber harassment detection scheme for Vine, a popular short video sharing platform, using a supervised classification scheme with features extracted from the owner profiles and video meta information [24]. Mariconti *et al.* developed an ensemble approach to proactively identify the YouTube videos where users are bullied by coordinated harassers with opposite political stands [25]. Barakat *et al.* developed a keyword spotting (KWS) approach to spot offensive words in the audio track of user video blogs [3]. Sutejo *et al.* proposed a deep learning-based long short-term memory (LSTM) framework to extract latent textual and acoustic features, and detect hate speech in videos [26]. While the above solutions can help track down a certain amount of hateful videos, they are also known to suffer from both false positives and false negatives in their detection results [5], [27]. Moreover, the absence of hateful content in the video itself is not sufficient to distinguish hatred-vulnerable videos from the hatred-free ones based on their definitions. To address the above limitations, we develop a content-agnostic scheme that is uniquely designed for detecting hatred-vulnerable videos by exploring the "wisdom" from the audience of the videos without the need to analyze the video content.

3. Problem Definition

In this section, we formally define the problem of detecting hatred-vulnerable videos from online video sharing platforms. We first define a few key terms that are used in our problem statement.

Definition 1. Video (V_i): a video instance V_i on online video sharing platforms typically includes the following components: i) a *title* that provides a brief textual description of the video content, ii) a *thumbnail* that gives the visual description of the video content, iii) *video content* is the actual video and audio content of the video, iv) a *description* that contains any additional information the video creator provides, v) *metadata* (e.g., number of views, number of likes), and vi) *comments* the video receives.

Definition 2. Hateful Content: the video content is considered as *hateful* if any part of the video content demonstrates or promotes violence or hatred against individuals or groups of people (e.g., race, group, religion)[2].

Definition 3. Hateful Comment: a user post in the comment section of the video is *hateful* if the comment contains any hateful speech against individuals or groups of people [1].

Definition 4. Hatred-vulnerable Video (labeled as "0"): a video is *hatred-vulnerable* if it receives only *hateful comments* but does not include any *hateful content*.

Definition 5. Hateful Video (labeled as "1"): a video is *hateful* if it contains *hateful content*.

Definition 6. Hatred-free Video (labeled as "2"): any video that is not in the *hatred-vulnerable* and *hateful* categories.

Using the key terms defined above, we define the online hatred-vulnerable video detection problem as a multi-class classification problem. In particular, given a set of N videos $\mathcal{V} = \{V_1, V_2, \cdots, V_N\}$ and the corresponding ground truth label $y_i \in \{0, 1, 2\}$ of each video $V_i \in \mathcal{V}$, the hatred-vulnerable video detection problem is to classify a video into one of the three classes: hatred-vulnerable ($y_i = 0$), hateful ($y_i = 1$) or hatred-free ($y_i = 2$). Let $\tilde{y_i}$ denotes the estimated label for video V_i, the problem is:

$$\arg\max_{\tilde{y_i}} P(\tilde{y_i} = y_i | V_i), \ \forall 1 \leq i \leq N. \tag{1}$$

4. Solution

In this section, we present the VulnerCheck scheme to solve the online hatred-vulnerable video detection problem defined above. The VulnerCheck is a supervised multi-class classification scheme that leverages a novel set of video content irrelevant features to effectively distinguish hatred-vulnerable, hateful, and hatred-free videos. These new features are extracted from three feature extraction modules. First, the *topological and semantic feature extraction (TSFE)* module is developed to learn features from the audience's interaction in a video's comments. Second, the *linguistic feature extraction (LFE)* module is designed to extract features from users' feedback by learning vector representation of the comments. Third, the *metadata feature extraction (MFE)* module collects a set of metadata features to support the detection of hatred-vulnerable videos. Finally, features extracted from these three modules are incorporated and fitted into a *supervised multi-class classification (SMC)* module to identify the hatred-vulnerable videos from all videos. An overview of VulnerCheck is shown in Figure 3.

[2]https://support.google.com/youtube/answer/2801939

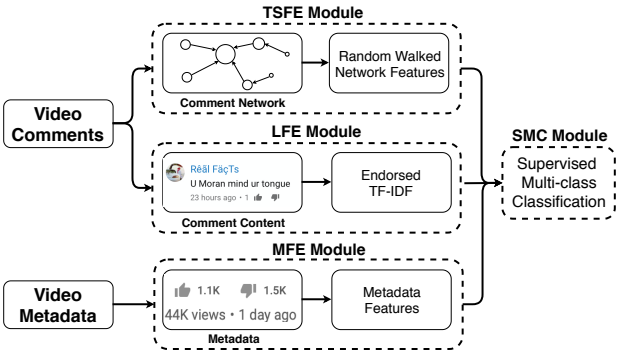


Figure 3: Overview of the VulnerCheck Scheme

4.1. Topological and Semantic Feature Extraction (TSFE)

The topological and semantic feature extraction module is developed to capture characteristics of the audience's discussion and their complex interactions in the comments of an online video. We observe that comment behavior of users vary across different video classes (i.e., hatred-vulnerable, hateful, and hatred-free). In particular, we identify two important categories of features of user comments: i) *topological* features including the network structure of the comment threads and their topological properties (e.g., length of the thread, replying direction), and ii) *semantic* features including the usage of hateful phrases in a comment, countering another hateful comment, and endorsement of other's comments. The VulnerCheck scheme effectively captures both topological and semantic features of online videos by devising two new mechanisms: i) we construct a novel semantic-aware comment network that captures both the semantics (level of hatred, counterspeech, and endorsement) and the topological structure of the networks; ii) we design a new embedding technique that effectively extracts useful features in the comment network. The details are presented below.

4.1.1. Comment Network Construction. First, we construct the comment network to capture the topological feature of the network. A typical comment structure of online video sharing platforms (e.g., YouTube) is in the form of comment threads. Each comment thread consists of a top-level comment and all sub-comments in reply to that comment. For each video, we define the comment network as a directed graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$, where $\mathbf{V}$ is a set of comment nodes, and $\mathbf{E}$ is a set of directed edges. Each edge $e_{v,v'} \in \mathbf{E}$ denotes a reply from comment v' to v. We also add a source node s to $\mathbf{V}$ to denote the video itself and create links between all top-level comments and node s.

4.1.2. Semantic Features. Second, we define semantic features from the constructed comment network. In particular, we focus on three types of semantic features that are observed to be characteristics of the hatred-vulnerable videos: *hate speech*, *counterspeech*, and *endorsement*.

The *hate speech* feature $f_h(v)$ is defined as the occurrence of hateful phrases in comment node v collected by Hatebase[3], the world's largest hate speech repository. In particular, $f_h(v)$ is "1" if a comment contains at least one hateful phrase, and "0" otherwise.

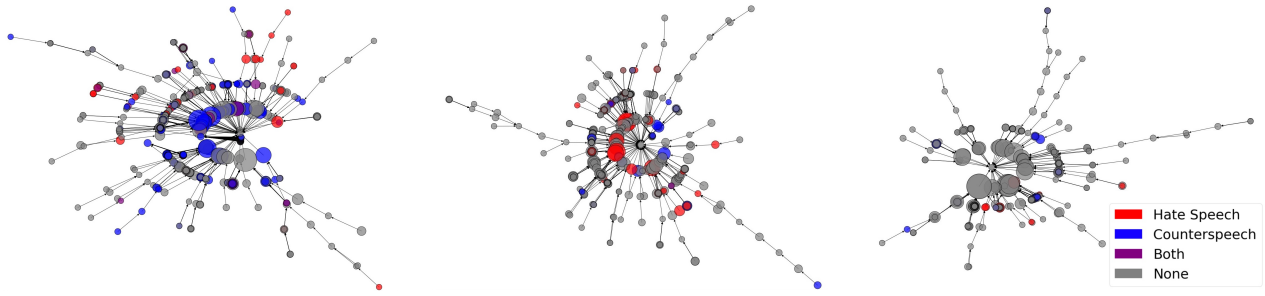
The *counterspeech* feature $f_c(v)$ is defined as the likelihood of comment node v containing a counterspeech. Counterspeech is recognized as the direct debunking response to a hateful comment [28]. For example, in a video reporting that Portland, Maine overrun with African migrants, a top-level comment "they are ILLEGAL ALIENS. Ship them back to their sh*t h*le countries" received a reply with counterspeech "when criticizing these people, remember where your origins are and who paid the heaviest price for your comfort today." Counterspeech is observed to be an alternative to combat the hateful or harmful speech without violating the normative of free speech [21]. We leverage the state-of-the-art pre-trained counterspeech detection tool developed by Mathew *et al.* [6] to identify counterspeech in the comments of each video. In particular, $f_c(v)$ is a probability that a comment contains a counterspeech.

The *endorsement* feature $f_e(v)$ is defined as the number of likes a comment node v receives. $f_e(v)$ is an indicator of positive feedback from users on each individual comment.

Figure 4 shows the hate speech, counterspeech, and endorsement features in the comment networks of three videos of different categories. We first observe that comments (e.g., especially the top-level ones) containing hateful phrases (i.e., red nodes in the figure) often receive more likes in hateful videos than in hatred-vulnerable videos. The reason is that the audience of hateful videos often share similar extreme ideology and are more likely to endorse a comment with hate speech. We also observe that hatred-free videos usually contain neutral topics and receive less hateful comments than the other two video categories. In contrast, we observe that counterspeech comments (i.e., blue nodes in the figure) in hatred-vulnerable videos receive more endorsements from the audience than the other two video categories. Such a phenomenon demonstrates that the comments of hatred-vulnerable videos appear to be more controversial than others and the hateful comments to a hatred-vulnerable video are more likely to be debunked by other users in the form of counterspeech. We also observe that hatred-free videos have fewer counterspeech comments due to the uncontroversial nature of their topics.

4.1.3. Topological and Semantic Features Extraction with Random Walk. After constructing the comment network with the above semantic features, we adopt the Random Walk (RW) algorithm [29] to jointly capture the topological and semantic features of the comment network [30]. In particular, the random walk process allows us to fuse the topological features (e.g., the depth of each traversal) and semantic features (e.g., hate speech, counterspeech, and endorsement) during each traversal. A random walk $RW(M, K)$ scheme randomly traverses a graph M times

[3]https://www.hatebase.org/



(a) Hatred-vulnerable (b) Hateful (c) Hatred-free

Figure 4: User Comment Network. The color of a node denotes hate speech feature of a comment: red - hateful only, blue - counterspeech only, purple - both hateful and counterspeech, grey - non-hateful and non-counterspeech. The size of a node represents the endorsement feature (i.e., the number of likes a comment receives). Note: for better visualization, we only keep threads with more than one comment in all network plots.

with at most K steps in each traversal [31], [32]. In particular, we randomly select a top-level comment node and employ the depth-first strategy to explore the comment network by recording the hate speech, counterspeech and endorsement features of each visited nodes along the random walk path. Formally, we define the random walk process for the aforementioned semantic features as follows:

Definition 7. Hate Speech Walk ($walk_h$): it is the network traversing process that traverses the directed graph $\mathbf{G}$ by randomly selecting a top-level comment node v_0 and choosing the direction of the next comment node in a depth-first fashion. In each step, we record the *hate speech feature* f_h of each visited comment node v_i along the path. Formally, the m^{th} walk $walk_h^m = \{walk_h^m(v_0), walk_h^m(v_1), \cdots, walk_h^m(v_{K-1})\}$, where $walk_h^m(v_i) = f_h(v_i) \forall i = 0, 1, \cdots, K-1$.

Definition 8. Counterspeech Walk ($walk_c$): it is the network traversing process that traverses the directed graph $\mathbf{G}$ by randomly selecting a top-level comment node v_0 and choosing the direction of the next comment node in a depth-first fashion. In each step, we record the *counterspeech feature* f_c of each visited comment node v_i along the path. Formally, the m^{th} walk $walk_c^m = \{walk_c^m(v_0), walk_c^m(v_1), \cdots, walk_c^m(v_{K-1})\}$, where $walk_c^m(v_i) = f_c(v_i) \forall i = 0, 1, \cdots, K-1$.

Definition 9. Endorsement Walk ($walk_e$): it is the network traversing process that traverses the directed graph $\mathbf{G}$ by randomly selecting a top-level comment node v_0 and choosing the direction of the next comment node in a depth-first fashion. In each step, we record the *hate speech feature* f_e of each visited comment node v_i along the path. Formally, the m^{th} walk $walk_e^m = \{walk_e^m(v_0), walk_e^m(v_1), \cdots, walk_e^m(v_{K-1})\}$, where $walk_e^m(v_i) = f_e(v_i) \forall i = 0, 1, \cdots, K-1$.

We perform the random walk process M times for each feature and limit the length of the path to be at most K steps. If a m^{th} traversal reaches the end node in less than K steps, the rest entries in $walk_h^m$, $walk_c^m$, $walk_e^m$ will be marked as non-hateful ($f_h = 0$), non-counterspeech ($f_c = 0$), and zero endorsement ($f_e = 0$), respectively. The extracted feature vectors are stored in feature matrices denoted as RW_h, RW_c, and RW_e for the corresponding semantic features. These matrices allow us to record not only the semantic feature of each comment node but also their corresponding topological properties simultaneously.

Finally, we perform the Principal Component Analysis (PCA) [33] to reduce the dimension of the feature vectors generated in the above random walk process. This is done to reduce the sparsity of the feature space and avoid the problem of overfitting in the final classification results [34]. We first apply PCA to extract the dominant patterns in each semantic feature matrix (i.e., RW_h, RW_c, and RW_e), denoted as $\mathbf{V}_h$, $\mathbf{V}_c$, and $\mathbf{V}_e$. Then, we concatenate the dominant patterns to represent the extracted topological and semantic features in the comment network as $\mathbf{V}_{network} = [\mathbf{V}_h, \mathbf{V}_c, \mathbf{V}_e]$.

4.2. Linguistic Feature Extraction (LFE)

We observe that the words used in the comments of different video classes are also different. Figure 5 shows the word clouds from the comments of hatred-vulnerable, hateful, and hatred-free videos. We note that the comments of both hatred-vulnerable and hateful videos contain a significant number of unwanted vocabularies (e.g., "a**hole", "fu*k"). However, the comments of hatred-vulnerable videos contain more neutral words to describe the potential hatred-sensitive video topics (e.g., "christian", "jews", "israel"). In contrast, the comments of hateful videos contain more extreme and biased words (e.g., "nigga", "blacks", "transgender"). We also observe that the common words (e.g., "earth", "video", "love") appear more often in the comments of hatred-free videos.

To capture the above differences regarding hatred-vulnerable, hateful, and hatred-free videos, we design a new lexical frequency feature - Endorsed Term Frequency-

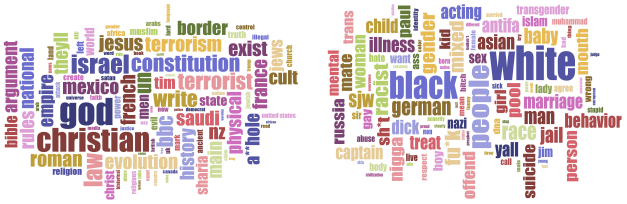
Inverse Document Frequency (ETF-IDF). The ETF-IDF is extended from the TF-IDF [35] which is a widely adopted measurement for the importance of terms in a set of documents. However, the vanilla TF-IDF treats each comment as an equal and ignores the popularity of the comments and the sensitivity of hateful vocabularies in the comments from user's feedback (e.g., endorsement). Therefore, the ETF-IDF jointly measures the term frequency and user feedback for each hatred-related term in a comment. In particular, we calculate the ETF-IDF of hatred-related terms from Hatebase as the product of its *TF-IDF* and the aggregated *number of likes* of comments that contain the corresponding hatred-related term. Formally, the ETF-IDF measure for a hateful phrase h in a document d is calculated as:

$$\text{ETF-IDF}_{h,d} = etf_{h,d} \cdot \log \frac{|D|}{1 + \sum_{d \in D} 1(etf_{h,d} > 0)} \quad (2)$$

where H is the corpus of hateful phrases and D is the set of documents where each document d contains all the comments of a video. $etf_{h,d}$ is the endorsed term frequency defined as:

$$etf_{h,d} = \frac{e_h \cdot f_{h,d}}{\sum_{h' \in H} f_{h',d}} \quad (3)$$

where e_h is the aggregated number of likes of comments containing the hateful phrase h. The ETF-IDF integrates users endorsement with the term frequency to compute the comprehensive hatred-related term importance measure. We then apply PCA to the ETF-IDF feature vector to further reduce it to a low dimensional feature vector, denoted as $\mathbf{V}_{linguistic}$, that is of a similar dimensionality as the comment network features (i.e., $\mathbf{V}_{network}$).



(a) Hatred-vulnerable (b) Hateful

(c) Hatred-Free

Figure 5: Word Cloud

4.3. Metadata Feature Extraction (MFE)

Additionally, we extract a set of complementary metadata features that are observed to be closely related to the identification of the hatred-vulnerable videos. In comparison to the latent feature extracted by the TSFE module, these metadata features are often more specific and intuitive. For example, we observe hatred-vulnerable videos appear more frequently in the news and education category while the hateful videos are more likely to appear in the comedy and entertainment category (because such videos often include musicals or animations to circumvent the censorship [5]). We empirically extracted 14 metadata features (denoted as $\mathbf{V}_{metadata}$) that describe both the video's metadata (e.g., number of views and likes) and its comment characteristics (e.g., number of words per comment, average length of a thread) for our model. A summary of the extracted metadata features is presented in Table 1. These metadata features are combined with features extracted from the TSFE and LFE modules, which will be used in a classifier for the final classification discussed next.

Table 1: Metadata Features

Feature	Description
`Comment Count`	Total # of comments for each video
`Dislike Count`	Total # of dislikes for each video
`Like Count`	Total # of likes for each video
`View Count`	Total # of views for each video
`Like to Dislike`	The ratio of like count to dislike count
`Daily View Count`	Avg. # of daily views for each video
`Like to View`	The ratio of like count to view count
`Duration`	Length of video in minutes
`Category`	Video Category
`Description URL Count`	Avg. # of URLs in the description
`Like Count per Comment`	Avg. # of likes for each comment
`Word Count per Comment`	Avg. # of words in each comment
`Length of Thread`	Avg. # of comments in threads with more than one comment
`Question Mark per Comment`	Avg. # of question marks in each comment
`Exclamation Mark per Comment`	Avg. # of exclamation marks in each comment

4.4. Supervised Multiclass Classification

Finally, we combine the comment network features ($\mathbf{V}_{network}$), linguistic features ($\mathbf{V}_{linguistic}$), and metadata ($\mathbf{V}_{metadata}$) features described above and apply the multiclass classification techniques to accurately identify hatred-vulnerable videos. In particular, we adopt a few state-of-the-art classification algorithms [36] and select the best-performing one as the classifier to integrate with the VulnerCheck scheme. The supervised classifiers include probabilistic classifiers (e.g., logistic regression), support vector machine, boosting and ensemble methods (e.g., XGBoost), and neural networks (e.g., multi-layer perceptron). The detailed performance evaluation for the selected set of classifiers and the VulnerCheck scheme is discussed in Section 6.

5. Data

In this section, we describe the collection, processing, and labeling process of the video dataset collected from YouTube for evaluation. YouTube is one of the largest video sharing platforms in the world, and more than 300 hours of

videos are uploaded to it every minute [4]. Considering the extremely large number of videos available on YouTube, it is challenging to directly collect and label hatred-related videos on YouTube. Alternatively, we start to collect video instances shared on a hate speech favorable platform, Gab [5], to overcome such a challenge. As an emerging social media platform that claims to promote free speech, Gab has been leveraged by users who were banned from mainstream social media to post and spread extreme and hateful content [37]. We also observe that video posts on Gab rely heavily on YouTube as a video library and many of the videos have the URLs from Youtube. To evaluate the performance of VulnerCheck, we crawl a set of Gab posts containing YouTube video links and record each video by its unique video ID. In particular, we collect a set of 1082 videos for our experiments. The publishing dates of the collected videos range from July 2006 to June 2019.

For each video, we collect the title, description, thumbnail, and comments information through YouTube API [6]. For the baselines that analyze the video content, we also collect the video and its corresponding transcript using the command-line video downloader youtube-dl [7]. We hire three independent human annotators to label the ground truth of each video by asking them to carefully watch the entire video clip and review the corresponding comments of the video. The annotators manually check the existence of both hateful content in the video clip and hateful comments in the comment section of the video before generating their labels. The ground truth labels are based on the majority vote of these human annotators. A summary of the dataset is reported in Table 2. We observe that the number of comments in each comment thread of the collected videos follows a long tail distribution (Figure 6). In particular, we observe that the majority of the comment threads have no more than 3 comments and thus set the length of the random walk process of our model to be 5 which will be discussed in Section 6.2. In addition, we translate all non-English comments to English using the Google Translation API [8].

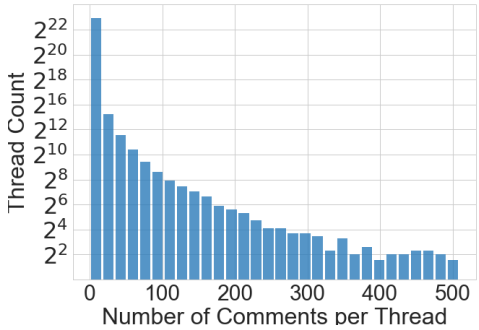


Figure 6: Distribution of Comments Count per Thread

[4]https://biographon.com/youtube-stats/
[5]https://gab.ai
[6]https://developers.google.com/youtube/v3/
[7]http://yt-dl.org
[8]https://cloud.google.com/translate/docs/

Table 2: Data Trace Statistics

Data Trace	Hatred-vulnerable	Hateful	Hatred-free
# of Videos	291	141	650
# of Comment Threads	449,200	349,636	1,130,542
Avg. Duration (minutes)	17.72	17.78	16.93
Avg. # of Comment Threads	1543.64	2479.69	1739.29
Comment Count	788,582	577,443	1,796,780
Avg. Comments Count	2709.90	4095.34	2764.28
Length of Thread	1.76	1.65	1.59
# of Distinct Users	378,926	312,853	1,079,071
# of Unique Words in the Comments	329,951	235,839	568,142
# of Words per Comment	16.45	14.63	20.50

6. Evaluation

In this section, we evaluate the performance of the VulnerCheck scheme on the real-world dataset described in Section 5. We compare the performance of VulnerCheck with the state-of-the-art baselines from the literature. The results show that the VulnerCheck scheme significantly outperforms all compared baseline methods by detecting hatred-vulnerable videos more accurately and efficiently.

6.1. Baselines

- **Universal Language Model Fine-tuning (ULMFiT) [38]:** it is a transfer learning method that can fine-tune text classification models with pre-trained natural language processing models. We take the comments of each video as the input to the ULMFiT framework and train a neural network classifier to detect the hatred-vulnerable videos.
- **Bag-of-Words (BoW) [39]:** it is a standard approach for text feature extraction. The hatred-vulnerable video classification task is performed by a multinomial Naïve Bayes classifier with the extracted bag-of-words features (e.g., the unigrams and bigrams in the comments of a video).
- **Sentiment-based Comment Network Embedding (SCNE) [40]:** it is a network embedding approach that extracts the sentiments of user comments of social media posts. In particular, the extracted features together with the metadata information are processed by the classifier to detect the hatred-vulnerable videos.
- **Content-based Recurrent Neural Network (RNN-GRU) [41]:** it is a recurrent neural network (RNN) based framework that learns vector-based document representation with pre-trained word embeddings (e.g., Global Vectors for Word Representation (GloVe) [42]). We combine the title, description, and transcript of the video as the inputs. The RNN with gated recurrent unit (GRU) is trained to accomplish the hatred-vulnerable video detection task.

Please note that we carefully tune the corresponding model parameters to achieve the best performance of the above baselines for a fair comparison with VulnerCheck.

6.2. Evaluation Metrics and Parameter Setting

To evaluate the performance of VulnerCheck in detecting the hatred-vulnerable videos, we use several widely adopted metrics for the multi-class classification task: *Accuracy*, *Precision*, *Recall*, and *F1 Score*. For *Precision*, *Recall*, and *F1 Score*, we calculate the macro-averaged score of these metrics. In addition, we also evaluate the detection performance on the metrics of *Cohan's Kappa Coefficient (Kappa)* and *Matthews Correlation Coefficient (MCC)* due to the imbalanced nature of the dataset. For all compared schemes, we use 70% of the dataset as the training set and perform 5-fold cross-validation on the training set to tune parameters. We set $M = 50$ and $K = 5$ for the random walk scheme for each semantic attribute. We set the maximum document frequency to be 0.6 in order to filter out common phrases that appear in the comments of all videos.

In an initial set of experiments, we integrate the features extracted by the VulnerCheck scheme with a few state-of-the-art supervised classifiers, including Logistic Regression (LR), Support Vector Machine (SVM), XGBoost, Random Forest (RF), Multi-layer Perceptron (MLP) as discussed in Section 4. We observe that XGBoost performs the best among all compared classifiers and select it as the classification module in the VulnerCheck scheme to identify the hatred-vulnerable videos.

6.3. Detection Accuracy

We first evaluate the detection accuracy of VulnerCheck and the compared baselines. The results are shown in Table 3. We observe that our VulnerCheck scheme consistently outperforms all baseline methods on all evaluation metrics. In particular, VulnerCheck achieves a performance gain of 6%, 4%, 8%, 11% in terms of accuracy compared to *ULMFiT*, *BoW*, *SCNE*, *RNN-GRU*, respectively. We also note that ULMFiT and BoW are prone to the misclassifications between hatred-vulnerable and hateful videos because they rely heavily on the accurate detection of hateful comments shared by those two video categories. SCNE has a sub-optimal performance due to a relatively large amount of negative sentiments embedded in the hateful comments from hatred-vulnerable and hateful videos, making the classification biased. The content-based approach (i.e., RNN-GRU) also fails to detect hatred-vulnerable videos effectively because it is not robust against hateful content creators who are sophisticated at crafting videos to circumvent content analysis tools. In contrast, the content-agnostic design of VulnerCheck leverages the intelligence of the audience by exploring a novel set of video content irrelevant features to effectively classify all video categories.

We also note that VulnerCheck achieves an improvement of 7% and 10% on the Kappa and MCC scores compared to the best-performing baseline (i.e., ULMFiT). Such a performance gain also demonstrates that the robustness of the VulnerCheck scheme over the imbalanced dataset. In addition, we present the Receiver Operating Characteristic (ROC) curve in Figure 7 to evaluate the robustness of all schemes with respect to the classification threshold. We observe that VulnerCheck continues to outperform all baselines when we tune the classification thresholds.

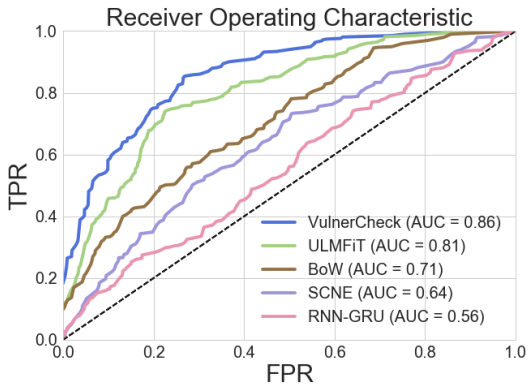


Figure 7: ROC Curve of All Compared Schemes

6.4. Influence of Training Size

In addition to the study of detection accuracy, we also investigate the robustness of the VulnerCheck scheme and all baseline methods with respect to the size of training data. In our experiment, we tune the training set size from 20% to 70% of the entire dataset and evaluate the classification performance (i.e., F1 score) of each method. The results are shown in Figure 8. We observe that the performance of the VulnerCheck scheme consistently improves as the amount of training data increases. VulnerCheck achieves the best performance across different sizes of training set. We also observe that the performance of the RNN-GRU baseline is stable but poor. This is because such a content-based approach primarily relies on the pre-trained word embeddings learned from the video content which might be misleading. The results on other metrics are similar and we omit them due to the space limit.

6.5. Detection Time

In the last set of experiments, we investigate the detection efficiency of the VulnerCheck scheme in terms of detection time. The detection time is defined as the amount of time a scheme takes to detect the hatred-vulnerable video after it has been originally published on YouTube. In particular, we tune the detection time from 10 minutes to 24 hours and only use the user comments received within the specified time window for all compared schemes. We measure the detection performance on the F1 score and the detection results are shown in Figure 9. We first observe that the performance of VulnerCheck continuously improves

Table 3: Classification Performance for All Methods

	Accuracy	Precision (Macro)	Recall (Macro)	F1 Score (Macro)	Kappa	MCC
XGBoost (VulnerCheck)	**0.7333**	**0.6973**	**0.5140**	**0.5169**	**0.4284**	**0.4465**
Logistic Regression (LR)	0.6351	0.3598	0.3668	0.3328	0.1009	0.1386
Support Vector Machine (SVM)	0.6526	0.3891	0.4194	0.3994	0.2324	0.2488
Random Forest (RF)	0.6702	0.4041	0.4206	0.4003	0.2387	0.2716
Multi-layer Perceptron (MLP)	0.5684	0.4745	0.4808	0.4537	0.2684	0.2882
ULMFiT	0.6796	0.5924	0.5071	0.5150	0.3607	0.3438
BoW	0.6982	0.4432	0.4355	0.4180	0.2894	0.3028
SCNE	0.6526	0.4347	0.3896	0.3694	0.1564	0.2061
RNN-GRU	0.6246	0.4705	0.3505	0.3062	0.0350	0.0659

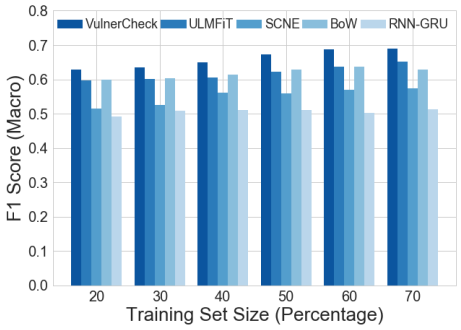


Figure 8: Training Set Size v.s. Performance

as the detection time increases (i.e., more input data become available). More importantly, VulnerCheck starts to outperform all baselines even at an early stage (e.g., within 10 minutes). It demonstrates the ability of VulnerCheck to detect hatred-vulnerable videos promptly after they are published. Such capability of VulnerCheck is critical for Youtube to take timely actions to prevent hateful comments from being spread on the detected videos.

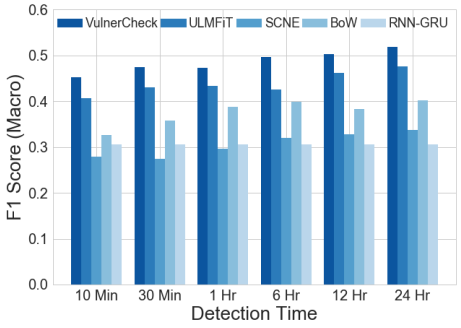


Figure 9: Detection Time v.s. Performance

7. Conclusion

In this paper, we develop VulnerCheck to address the problem of online hatred-vulnerable video detection. Our scheme leverages the wisdom of the audience by exploring a set of novel features that encode their attitudes and interactions into a semantic-aware user comment network. VulnerCheck does not depend on the analysis of the video content and is hence robust against the sophisticated content creators on the video sharing platforms. We evaluate our scheme using a real-world dataset collected from YouTube. The results demonstrate that our scheme outperforms state-of-the-art baselines by identifying the hatred-vulnerable videos in a more effective and efficient manner. We believe VulnerCheck is the first step towards a new path to reliably fight against the hateful content while protecting the freedom of speech on online video sharing platforms.

Acknowledgement

This research is supported in part by the National Science Foundation under Grant No. CNS-1845639, CNS-1831669, Army Research Office under Grant W911NF-17-1-0409. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Office or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

[1] H. M. Saleem, K. P. Dillon, S. Benesch, and D. Ruths, "A web of hate: Tackling hateful speech in online social spaces," *arXiv preprint arXiv:1709.10159*, 2017.

[2] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, and L. Kaplan, "The age of social sensing," *Computer*, vol. 52, no. 1, pp. 36–45, 2019.

[3] M. Barakat, C. Ritz, and D. A. Stirling, "Detecting offensive user video blogs: An adaptive keyword spotting approach," in *2012 International Conference on Audio, Language and Image Processing*. IEEE, 2012, pp. 419–425.

[4] "Youtube bans hateful videos from platform," https://www.wsj.com/articles/youtube-bans-supremacist-videos-11559754035, accessed: 2019-08-05.

[5] "Youtube's new moderators mistakenly pull right-wing channels," https://fortune.com/2018/02/28/youtube-right-wing-channels/, accessed: 2019-03-14.

[6] B. Mathew, P. Saha, H. Tharad, S. Rajgaria, P. Singhania, S. K. Maity, P. Goyal, and A. Mukherjee, "Thou shalt not hate: Countering online hate speech," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 13, no. 01, 2019, pp. 369–380.

[7] M. H. Ribeiro, P. H. Calais, Y. A. Santos, V. A. Almeida, and W. Meira Jr, "Characterizing and detecting hateful users on twitter," in *12th International AAAI Conference on Web and Social Media*, 2018.

[8] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan, "Youtube-8m: A large-scale video classification benchmark," *arXiv preprint arXiv:1609.08675*, 2016.

[9] Z. Waseem and D. Hovy, "Hateful symbols or hateful people? predictive features for hate speech detection on twitter," in *Proceedings of the NAACL student research workshop*, 2016, pp. 88–93.

[10] D. Wang, T. Abdelzaher, L. Kaplan, R. Ganti, S. Hu, and H. Liu, "Reliable social sensing with physical constraints: analytic bounds and performance evaluation," *Real-Time Systems*, vol. 51, no. 6, pp. 724–762, 2015.

[11] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 6645–6649.

[12] D. Y. Zhang, L. Song, Q. Li, Y. Zhang, and D. Wang, "Streamguard: A bayesian network approach to copyright infringement detection problem in large-scale live video sharing systems," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018.

[13] "The difference between youtube's automatic captions and a video captions service," https://www.rev.com/blog/video-captions-different-methods, accessed: 2019-08-13.

[14] T. Davidson, D. Warmsley, M. Macy, and I. Weber, "Automated hate speech detection and the problem of offensive language," in *Eleventh International AAAI Conference on Web and Social Media*, 2017.

[15] D. Wang, T. Abdelzaher, and L. Kaplan, *Social sensing: building reliable systems on unreliable data*. Morgan Kaufmann, 2015.

[16] D. Wang, T. Abdelzaher, L. Kaplan, and C. C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 2013, pp. 530–539.

[17] J. Waldron, *The harm in hate speech*. Harvard University Press, 2012.

[18] D. Wang, M. T. Amin, S. Li, T. Abdelzaher, L. Kaplan, S. Gu, C. Pan, H. Liu, C. C. Aggarwal, R. Ganti *et al.*, "Using humans as sensors: an estimation-theoretic perspective," in *Information Processing in Sensor Networks, IPSN-14 Proceedings of the 13th International Symposium on*. IEEE, 2014, pp. 35–46.

[19] P. Badjatiya, S. Gupta, M. Gupta, and V. Varma, "Deep learning for hate speech detection in tweets," in *Proceedings of the 26th International Conference on World Wide Web Companion*, 2017.

[20] J. Bartlett and A. Krasodomski-Jones, "Counter-speech examining content that challenges extremism online," *DEMOS, October*, 2015.

[21] I. Gagliardone, D. Gal, T. Alves, and G. Martinez, *Countering online hate speech*. Unesco Publishing, 2015.

[22] C. Schieb and M. Preuss, "Governing hate speech by means of counterspeech on facebook," in *66th ica annual conference, at fukuoka, japan*, 2016, pp. 1–23.

[23] B. Mathew, N. Kumar, P. Goyal, A. Mukherjee *et al.*, "Analyzing the hate and counter speech accounts on twitter," *arXiv preprint arXiv:1812.02712*, 2018.

[24] R. I. Rafiq, H. Hosseinmardi, R. Han, Q. Lv, S. Mishra, and S. A. Mattson, "Careful what you share in six seconds: Detecting cyberbullying instances in vine," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*. ACM, 2015, pp. 617–622.

[25] E. Mariconti, G. Suarez-Tangil, J. Blackburn, E. De Cristofaro, N. Kourtellis, I. Leontiadis, J. L. Serrano, and G. Stringhini, ""you know what to do": Proactive detection of youtube videos targeted by coordinated hate attacks," *arXiv preprint arXiv:1805.08168*, 2018.

[26] T. L. Sutejo and D. P. Lestari, "Indonesia hate speech detection using deep learning," in *2018 International Conference on Asian Language Processing (IALP)*. IEEE, 2018, pp. 39–43.

[27] D. Wang, L. Kaplan, T. Abdelzaher, and C. C. Aggarwal, "On credibility estimation tradeoffs in assured social sensing," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 6, pp. 1026–1037, 2013.

[28] L. Wright, D. Ruths, K. P. Dillon, H. M. Saleem, and S. Benesch, "Vectors for counterspeech on twitter," in *Proceedings of the First Workshop on Abusive Language Online*, 2017, pp. 57–62.

[29] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2016.

[30] D. Y. Zhang, L. Shang, B. Geng, S. Lai, K. Li, H. Zhu, M. T. Amin, and D. Wang, "Fauxbuster: A content-free fauxtography detector using social media comments," in *2018 IEEE International Conference on Big Data (Big Data)*. IEEE, 2018, pp. 891–900.

[31] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: Online learning of social representations," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 701–710.

[32] L. Shang, D. Y. Zhang, M. Wang, S. Lai, and D. Wang, "Towards reliable online clickbait video detection: A content-agnostic approach," *Knowledge-Based Systems*, vol. 182, p. 104851, 2019.

[33] I. Jolliffe, "Principal component analysis," in *International encyclopedia of statistical science*. Springer, 2011.

[34] D. Y. Zhang, B. Ni, Q. Zhi, T. Plummer, Q. Li, H. Zheng, Q. Zeng, Y. Zhang, and D. Wang, "Through the eyes of a poet: Classical poetry recommendation with visual input on social media," 2019.

[35] S. Robertson, "Understanding inverse document frequency: on theoretical arguments for idf," *Journal of documentation*, vol. 60, no. 5, pp. 503–520, 2004.

[36] E. Alpaydin, *Introduction to machine learning*. MIT press, 2014.

[37] L. Lima, J. C. Reis, P. Melo, F. Murai, L. Araujo, P. Vikatos, and F. Benevenuto, "Inside the right-leaning echo chambers: Characterizing gab, an unmoderated social system," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 515–522.

[38] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," *arXiv preprint arXiv:1801.06146*, 2018.

[39] T. Joachims, "Text categorization with support vector machines: Learning with many relevant features," in *European conference on machine learning*. Springer, 1998, pp. 137–142.

[40] A. Zadeh, M. Chen, S. Poria, E. Cambria, and L.-P. Morency, "Tensor fusion network for multimodal sentiment analysis," *arXiv preprint arXiv:1707.07250*, 2017.

[41] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

[42] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 2014, pp. 1532–1543.