StreamGuard: A Bayesian Network Approach to Copyright Infringement Detection Problem in Large-scale Live Video Sharing Systems

Daniel (Yue) Zhang, Lixing Song, Qi Li, Yang Zhang, Dong Wang
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN, USA
{*yzhang40, lsong2, qli8, yzhang42, dwang5*}*@nd.edu*

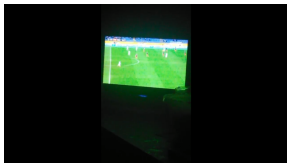
***Abstract*—Copyright infringement detection is a critical problem in large-scale online video sharing systems: the copyright-infringing videos must be correctly identified and removed from the system to protect the copyright of the content owners. This paper focuses on a challenging problem of detecting copyright infringement in *live* video streams. The problem is particularly difficult because i) streamers can be sophisticated and modify the title or tweak the presentation of the video to bypass the detection system; ii) legal videos and copyright-infringing ones may have very similar visual content and descriptions. We found current commercial copyright detection systems did not address this problem well: a large amount of copyrighted content bypasses the detection system while legal streams are taken down by mistake. In this paper, we develop the StreamGuard, an unsupervised Bayesian network based copyright infringement detection system that addresses the above challenges by leveraging *live chat messages* from the audience. We evaluate StreamGuard on real-world live video streams collected from YouTube. The results show that StreamGuard is effective and efficient in identifying the copyright-infringing videos.**

I. Introduction

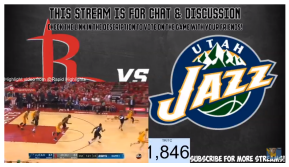
The advent of large-scale online video sharing platforms such as YouTube and Twitch has offered grassroots users the ability to broadcast and watch live videos on a global scale. Different from traditional static video content, live video streams are generated and consumed in real-time. A critical problem of the online video sharing system is the *copyright infringement issue* where users can stream and watch copyrighted live events such as sports matches and TV shows, without the authorization of content owners [1], [2]. The copyright infringement issue, if not addressed appropriately, can negatively impact the video sharing system by recommending copyright-infringing video streams or showing those videos in the search results to millions of audience, causing a huge financial loss to the content owners.

The video sharing platforms have made many efforts to detect copyright-infringing videos. YouTube, for example, has developed a proprietary copyright protection system called ContentID [3]. ContentID compares each uploaded video against a database of copyrighted video files to check for unauthorized content. ContentID also allows the content owners to manually file reports against the copyright-infringing videos that they have identified [3]. However, ContentID has been identified to perform poorly in detecting unauthorized streams due to two critical challenges: 1) the database cannot be created for live video streams given the fact the streams are generated in real-time [4]; 2) the content owners may not be able to identify all copyright-infringing videos due to the excessive manual labor involved [2]. In fact, YouTube has been criticized for failing to detect a large body of copyrighted contents while falsely taking down legal streams uploaded by streamers [1].

Besides the commercial solutions, existing mainstream copyright protection techniques such as fingerprinting [5] and watermarking [6], focus on static content (e.g., digital music, software, or ebooks) and cannot be directly applied to copyright detection on live video streams that are generated in *real-time* [7]. Alternatively, several tools have been recently developed to detect the similarities of videos [8], which can be potentially used to detect copyrighted video contents. However, they cannot address the critical challenge in our problem in that sophisticated streamers in video sharing platforms can manipulate the way the video is presented, making it appear to be hardly distinguishable from the original content (See Figures 1).



(a) Embedded TV Screen



(b) Video with Camouflage

Figure 1: Copyright-infringing videos uploaded by sophisticated streamers that have bypassed the ContentID of YouTube

In this paper, we propose a new *StreamGuard* scheme to effectively identify copyright-infringing live video streams

[1]https://www.engadget.com/2017/07/28/youtube-illegal-livestreams/

(see Figure 2). Inspired by the recent advances in the social sensing application paradigm [9]–[13], we leverage the "sensing data" (i.e., chat messages) provided by the audience to detect the copyright infringements. We observed that live chat messages often reveal important information about the copyright infringement of the video. For example, if the audience of a live soccer match is reminding the streamer to change the title or modify the video description, it is very likely they are collusively trying to bypass the detection system and the video is unauthorized. In StreamGuard, we develop a Bayesian network based latent semantic analysis framework that estimates the copyright infringement label of a video by exploring the live chat messages as well as the chatting patterns of the audience of the video. It is worth mentioning that a crowdsourcing-based scheme (referred to as CCID) has been recently proposed to solve the copyright infringement detection problem in live video streams [2]. However, CCID is a supervised approach and require a significant amount of manually labeled data to train its detection model. In contrast, the proposed StreamGuard scheme is designed to be *unsupervised*, which is motivated by the observation that well-annotated live video stream datasets for training can be prohibitively expensive or impractical for fresh live stream contents (e.g., a new TV show) [14].

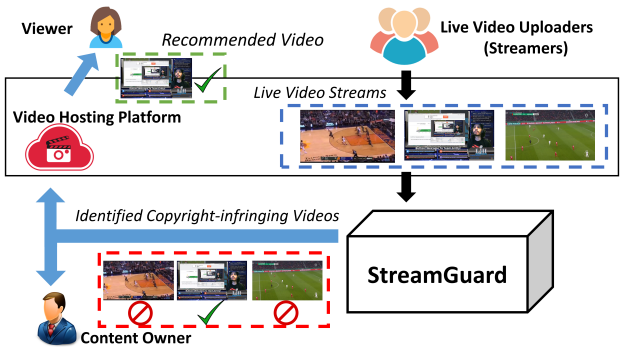


Figure 2: Overview of StreamGuard System

To our knowledge, StreamGuard is the first unsupervised solution to address the copyright infringement issue for live video streams. The proposed methods in StreamGuard is content-free (i.e., does not rely on the visual content of the videos). Therefore, it is robust against streamers who would intentionally change the presentation of the video. Additionally, StreamGuard performs the detection tasks *without accessing* the original copyrighted content or training data, which alleviates the need of well-annotated live video stream datasets that can be prohibitively expensive or impossible (e.g., fresh live stream contents) to obtain in practice. Preliminary results on two live stream video datasets collected from YouTube show StreamGuard is significantly more effective than baselines.

II. Problem Statement

In this section, we present the copyright infringement detection problem in live video streams. We assume that a video sharing platform hosts a set of candidate live videos $\mathcal{V} = \{v_1^\omega, v_2^\omega ... v_V^\omega\}$ related to a piece of copyrighted content (e.g., a new TV episode) $\omega, 1 \leq \omega \leq \Omega$. For the ease of notation, we just focus on one piece of the copyrighted content and omit the superscript (i.e., ω) in the rest of the paper. Each video $v \in \mathcal{V}$ is associated with a 5-tuple, i.e., $v = (t_v^{start}, t_v^{end}, \mathcal{C}_v, L_v)$ where t_v^{start} and t_v^{end} refer to the timestamp when the video starts and ends, respectively. Each video contains a set of live chat messages $\mathcal{C}_v$ (see Figure 3). We assume there exist a total of U users who chat on live videos on the platform: $\mathcal{U} = \{u_1, u_2, ..., u_U\}$. $\mathcal{C}_{v,u}$ denotes the chat messages posted by user u about video v. Each video also has a ground truth label L_v. We label a video as *"True" (i.e., Copyright-Infringing)* if it contains the actual copyrighted content (e.g., broadcasting a live World Cup match, streaming the latest episode of "West World"). We label a video as *"False" (i.e., Non-copyright Infringing)* if it does not contain any copyrighted content.

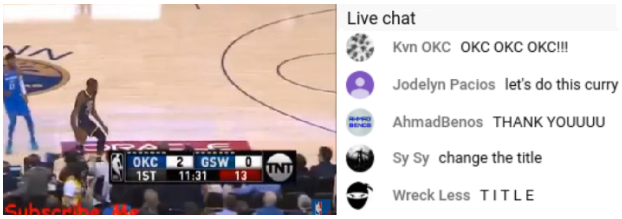


Figure 3: An Example of a Copyright-infringing Video Stream with Live Chat on YouTube

We summarize the assumptions of our model as follows.

- *Transient Content:* the copyrighted video stream is assumed to be generated and consumed in real-time and its content cannot be obtained in advance.
- *Sophisticated Streamers:* we assume the stream uploaders can intentionally modify the video's content and description to avoid being detected by the video sharing platform (see Figure 1).
- *Lack of Training Data:* we assume that copyrighted content can be fresh (e.g., a new TV show) and there might be no training data available for the copyright detection system.

The goal of StreamGuard is to identify the copyright-infringing videos and report them to legal content owners and the video hosting platforms by exploring the live chat messages of the audience. Formally, for each video stream, we want to uncover its ground truth label (copyright-infringing or not), i.e.,

$$\arg\max_{\tilde{L_v}} Pr(\tilde{L_v} = L_v | \mathcal{C}_v), \ \forall v \in \mathcal{V} \quad (1)$$

where $\tilde{L_v}$ denotes the estimated label for video v and $\mathcal{C}_v$ is the set of chat messages of v.

III. Solution: The StreamGuard System

In this section, we present the StreamGuard system for copyright infringement detection of live video streams. StreamGuard consists of three main components: 1) a Live Stream Crawler (LSC) module that collects live chat of videos in real-time; 2) a Semantic Feature Extraction (SFE) module that extracts the indicative hints from the live chat messages of the videos; 3) a Latent Semantic-aware Copyright Detection (LSCD) module that jointly models the users' latent chatting patterns and the labels of the videos using a Bayesian Latent Semantic Analysis approach.

A. Live Stream Crawler (LSC) for Data Collection

We first describe the Live Stream Crawler (LSC) module that is designed to crawl live video streams from YouTube. We developed a distributed live stream crawling system using Selenium and Docker. The system consists of a local master node that crawls the Internet to collect the schedules for live events. For example, we crawl FOX Sports [2] to get a list of scheduled soccer events. The master node then kicks off the video crawling jobs at the beginning of the scheduled event. The actual data crawling jobs are performed on a set of virtual machines instances at Amazon Web Service.

For each video, the LSC system collects the real-time chat messages from the audience and screenshots of the live video stream (captured every 30 seconds). The collected screenshots are to obtain the ground-truth label for each video. We describe the labeling process in Section IV.

We also crawl a *terminology dictionary* that consists of keywords (e.g., the team names and terminologies used in a sporting event) that are related to the copyrighted content to be protected. Such a dictionary is used to analyze the relevance of the chat messages to the copyrighted content (discussed in the next subsection).

B. Semantic Feature Extraction (SFE) from Live Chats

In the collected video datasets, we observe that a significant amount of chat messages of the videos actually contain valuable "hints" on whether the video is copyright-infringing or not. In StreamGuard, we focus on two types of features of a chat message: the *observation score* and *emotion score*. The observation score is defined below:

DEFINITION 1. Observation Score: an observation score of a chat message is an integer value that represents the extent to which the chat messages indicates a video is copyright-infringing. More specifically, the observation score is derived based on four different indicators of the chat messages defined as follows.

- *Colluding Behavior Indicator (ρ_{col}) :* the terms in a chat message that indicate that the audience is colluding with the streamer to bypass the copyright detection. Examples include terms such as "change the title" and "change the description".
- *Content Relevance Indicator (ρ_{rel}):* the terms in a chat message that are directly relevant to the content of the event. Examples include the team names of a World Cup game and the names of the actors/actresses in a TV show.
- *Video Quality Indicator (ρ_{qua}):* the terms in a chat message related to the quality of the video (e.g., "laggy", "full screen", "sound"). We observe that the audience tend to care more about the quality of a video if the video contains the copyrighted content that they expect to watch.
- *Debunking Indicator (ρ_{deb}):* the terms in a chat message that indicates direct criticisms of the video content (e.g., "fake, not working, go to my stream instead") of a video.

Table I shows some examples of the above indicators (in bold text) from our collected video datasets. For each type of indicator, we define a *indicator score* of a chat message as the number of terms of the indicators. The keywords/terms for the content relevance indicator are defined in the terminology dictionary (Section III-A). For other indicators, we define their relevant keywords/terms based on the prior knowledge from historic chat messages we collected. For example, a chat message "leave it full screen just change title. Let's go Cavs!" has an ρ_{qua} score of 2 and ρ_{rel} score of 1 since it contains two terms (i.e., "full screen" and "change title") that match the keywords of the video quality indicator and one term (i.e., "Cavs") related to the content relevance indicator.

For each chat message, an observation score (O) is defined as an aggregation of above indicator scores:

$$O = \begin{cases} \rho_{col} + \rho_{rel} + \rho_{qua}, & \rho_{deb} = 0 \\ 0, & \rho_{deb} > 0 \end{cases} \quad (2)$$

The intuition of the observation score is: the higher the score is, the more likely the corresponding message is copyright-infringing. If a chat message contains a debunking indicator, we set the observation score as 0 to indicate the video is unlikely to be copyright-infringing.

We also observe the emotion expressed in a chat message is related to the copyright infringement of the video content. For example, users may express happiness and excitement when they find a copyright-infringing video online. On the contrary, users often post curses and negative comments when they figure out a video is actually fake (i.e., non-copyright-infringing). We define the emotion score of a chat message as follows.

DEFINITION 2. Emotion Score: the polarity of sentiment

[2]https://www.foxsports.com/soccer/schedule

Table I: Examples of Indicators

Composition of Observation Indicators	Example Chat Messages (Bold Indicates Matched Terms)
Colluding Behavior	I'd recommend **not showing the live** during breaks so you don't get taken down **Change the title** and don't get greedy for viewers
Content Relevance	**Isaiah** needs to come back to the **line up**. **Cavs** cannot just rely on **LeBron James** who **scored** the first **basket** for **Rockets**?
Video Quality	why are u torturing us with bad **clarity**, **angle** and **no sound**? **laggy** but still appreciated
Debunking	**FAKE** DO NOT BOTHER bruh its weird af **wont work**

expressed by a chat message. The emotion score is assumed to be *categorical* - positive, neutral, and negative.

The emotion score is derived using the TextBlob's polarity analysis tool [15], which is a state-of-the-art sentiment analyzer. For example, we treat the emotion score as "positive" if polarity $> \theta$, "negative" if polarity $< -\theta$, and "neutral" if $-\theta \leq$ polarity $\leq \theta$. We found $\theta = 0.2$ turns out to be a reasonable value from our experiment.

To further illustrate the intuition of picking observation and emotion scores as our semantic features, we plot the score distribution in one of our collected datasets (Figure 4). We observe that the observation and emotion scores in copyright-infringing videos are clearly higher than the ones in non-copyright-infringing videos. This observation validates the chosen semantic features can be potentially important in copyright infringement detection.

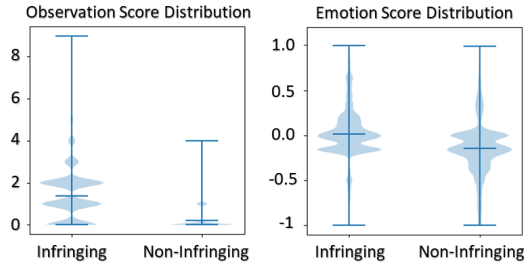


Figure 4: Violin Plot of Observation and Emotion Score Distribution in the Soccer Dataset

C. Latent Semantic-aware Copyright Detection (LSCD)

In this subsection, we present the Latent Semantic-aware Copyright Detection (LSCD) module to estimate if a video is copyright-infringing using the semantic features extracted from the chat messages as we discussed above. The LSCD module is designed to be *unsupervised* by considering the fact that well-annotated live video streams are prohibitively expensive or impossible to obtain (e.g., fresh live stream contents) in practice [16].

1) Model Intuition: In the previous subsection, we observe that both observation and emotion scores are potentially good features to differentiate copyright-infringing videos from non-copyright-infringing ones. However, a simple aggregation of those scores may lead to unsatisfactory detection results (shown in Section IV). Table II shows an illustrative example extracted from one of our datasets. In this example, Alice and Bob comment on two video streams related to the live broadcast of an NBA game (one is copyright-infringing (true) and one is not (false)). The aggregated observation and emotion scores from Alice and Bob on two videos are the exact same. It is challenging to identify which video is copyright-infringing by directly using the aggregated scores.

To address this problem, we develop a principled LSCD model. The key idea of the LSCD model is to identify the latent chatting behavior pattern of a user and its relationship to the copyright infringement of the chatted video. For example, if we know that Alice tends to post negative messages on copyright-infringing videos (e.g., complaining about video quality) and Bob tends to post negative comments on non-copyright-infringing videos (e.g., complaining about the relevance), we can easily distinguish the two videos in the above example. In the next subsection, we discuss how LSCD models the latent user chatting patterns in detail.

2) Model Details: We first introduce key variables of the LSCD model and their generation process. In LSCD, the *observed* variables are the observation scores and the emotion scores of the chat messages. The *latent* variables are the chatting patterns of users and the copyright infringement labels of the videos. The key notations of LSCD are summarized in Table III. The underlying generation process of these parameters is discussed below.

1) User Chatting Pattern (Latent): We assume a set of K latent user chatting patterns in LSCD. Examples of such patterns can be "posting chat messages with high observation scores and positive emotions in copyright-infringing videos" and "posting messages with low observation scores and neutral emotions in non-copyright-infringing ones".

Let $Z_u \in \{1, 2...K\}, u \in U$ denote the latent chatting pattern of user u. For the ease of notation, we use l to represent the video label L_v and k to represent Z_u. For each user, the latent user chatting pattern is assumed to be generated from a Multinomial distribution with parameter

Table II: Examples of User Chatting Patterns on Two NBA Streams

Video Id	Video Label	Chat Message (Bold Indicates Matched Terms)	Observation Score	Emotion Score
1	True (Copyright-Infringing)	Alice: Is anyone else video skipping. Supper **Laggy**!	1 (Video Quality)	Negative
		Bob: Hey vanila are u talking to me?	0	Neutral
2	False (Non-Copyright-Infringing)	Alice: anyone know a site?	0	Neutral
		Bob: Stop putting bull sh*t up I'm just trying to watch the **raptors** take a W.	1 (Content Relevance)	Negative

Table III: Definition and Notation

$\mathcal{C}_{v,u}$	chat messages posted by user u in video v
$O_{v,u}$	The observation scores of $\mathcal{C}_{v,u}$
$E_{v,u}$	The emotion scores of $\mathcal{C}_{v,u}$
L_v	The copyright infringement label of v
Z_u	The latent chatting pattern of user u
K	total number of latent chatting patterns
l	shorthand notation for L_v, $l = 0$ *or* 1
k	shorthand notation for Z_u, $1 \leq k \leq K$

$\theta^{(Z)}$:

$$k \sim Multinomial(\theta^{(Z)})$$

where $\theta^{(Z)}$ is the chatting pattern prior generated from a Dirichlet distribution with hyperparameter $\alpha^{(Z)}$:

$$\theta^{(Z)} \sim Dirichlet(\alpha^{(Z)})$$

Similar to the LDA model [17], the hyperparameter $\alpha^{(Z)}$ controls the "density" of the latent chatting patterns: the higher the value of $\alpha^{(Z)}$ is, the more user chatting patterns would appear in a video's chat messages.

2) Label of Live video Streams (Latent): For each video, we generate the binary latent label of the video from a Bernoulli Distribution:

$$l \sim Bernoulli(\phi_v)$$

where ϕ_v is the prior label distribution generated from a Beta distribution with hyperparameters $\beta = [\beta_0, \beta_1]$:

$$\phi_v \sim Beta(\beta_0, \beta_1)$$

The hyperparameter β_1 governs the probability of a video being copyright-infringing. In practice, if we do not have prior knowledge on the distribution of copyright-infringing videos v.s. non-copyright-infringing ones, we set $\beta_0 = \beta_1$.

3) Emotions and Observations (Observed Variables)

In our LSCD model, we assume the semantic features of chatting messages (i.e., observation and emotion scores) of a video are governed by both the latent label of the video and the latent user chatting behavior of the user. In particular, given the latent label of videos l and the latent chatting patterns k, the observation scores (denoted as $O_{v,u}$) of the chat messages posted by user u in video v is sampled from a Poisson distribution with parameter $\lambda_{l,k}$:

$$O_{v,u} \sim Lambda^{(O)}(\lambda_{l,k})$$

where the prior $\lambda_{l,k}$ is generated from a Gamma distribution with hyperparameters $\alpha^{(O)}_{l,k}, \beta^{(O)}_{l,k}$:

$$\lambda_{l,k} \sim Gamma^{(O)}(\alpha^{(O)}_{l,k}, \beta^{(O)}_{l,k})$$

We chose Poisson distribution based on two observations on the collected live chat messages: 1) the observation score is a discrete value; 2) the frequency of the score first increases as the observation score increases and immediately drops when the score becomes large (see Figure 4) which matches the characteristic of a Poisson distribution.

The categorical emotions scores $E_{v,u}$ of the chat messages posted by user u in video v is sampled from a Multinomial distribution with parameter $\theta^{(E)}_{l,k}$:

$$E_{v,u} \sim Multinomial(\theta^{(E)}_{l,k})$$

where the prior $\theta^{(E)}_{l,k}$ is generated from a Dirichlet distribution with hyperparameter $\alpha^{(E)}_{l,k}$:

$$\theta_{l,k} \sim Dirichlet(\alpha^{(E)}_{l,k})$$

Figure 5 shows the plate graph of the LSCD model. In the graph, Θ_0 is a vector of parameters given the video label is "False": $\Theta_0 = [\theta^{(E)}_{0,1}, ..., \theta^{(E)}_{0,K}, \lambda^{(O)}_{0,1}, ..., \lambda^{(O)}_{0,K}]$. α_0 is a vector of priors that governs Θ_0: $\alpha_0 = [\alpha^{(O)}_{0,1}, ..., \alpha^{(O)}_{0,K}, \beta^{(O)}_{0,1}, ..., \beta^{(O)}_{0,K}, \alpha^{(E)}_{0,1}, ..., \alpha^{(E)}_{0,K}]$. Similarly, Θ_1 is a vector of parameters given the video label is "True": $\Theta_1 = [\theta^{(E)}_{1,1}, ..., \theta^{(E)}_{1,K}, \lambda^{(O)}_{1,1}, ..., \lambda^{(O)}_{1,K}]$. α_1 is a vector of priors that governs Θ_1: $\alpha_1 = [\alpha^{(O)}_{1,1}, ..., \alpha^{(O)}_{1,K}, \beta^{(O)}_{1,1}, ..., \beta^{(O)}_{1,K}, \alpha^{(E)}_{1,1}, ..., \alpha^{(E)}_{1,K}]$.

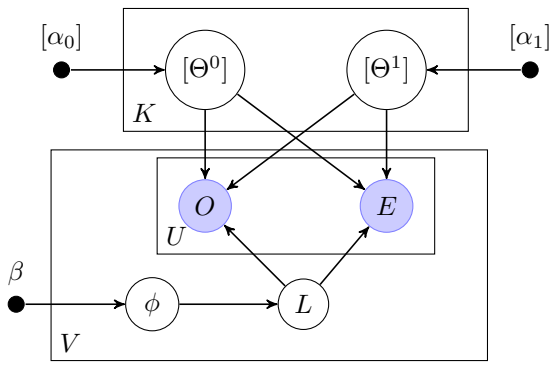


Figure 5: Plate Diagram of LSCD

3) Parameter Inference via Gibbs Sampling: In LSCD, the likelihood function of the observations, latent variables and hidden parameters given the hyperparameters $\boldsymbol{\alpha_0}$,$\boldsymbol{\alpha_1}$, and $\boldsymbol{\beta}$ is derived as:

$$p(\boldsymbol{E},\boldsymbol{O},\boldsymbol{L},\boldsymbol{Z},\boldsymbol{\phi},\boldsymbol{\Theta^0},\boldsymbol{\Theta^1}|\boldsymbol{\alpha_0},\boldsymbol{\alpha_1},\boldsymbol{\beta}) = \prod_{k\in K}\prod_{l\in\{0,1\}}\Big[p(\theta_{l,k}^{(E)}|\alpha_{l,k}^{(E)}) \; p(\lambda_{l,k}^{(O)}|\alpha_{l,k}^{(O)})\Big] \times \prod_{v\in\mathcal{V}}\Bigg\{p(\phi_v|\beta)\sum_{l\in\{0,1\}}p(l|\phi_v)^l(1-p(l|\phi_v))^{1-l} \prod_{u\in\mathcal{U}}\Big[\sum_{k\in K}p(E_{v,u}|\Theta_{l,k}^{(E)})p(O_{v,u}|\lambda_{l,k}^{(O)})\Big]\Bigg\} \tag{3}$$

Given the above likelihood function, we can jointly estimate the latent label of the video and the chatting behavior pattern of the users by deriving the *maximum a posterior (MAP)* estimate:

$$(\tilde{\boldsymbol{L}},\boldsymbol{Z})_{MAP} = \arg\max_{\boldsymbol{L},\boldsymbol{Z}} \iiint P(\boldsymbol{E},\boldsymbol{O},\boldsymbol{L},\boldsymbol{Z},\boldsymbol{\Theta^0},\boldsymbol{\Theta^1},\phi)d\phi d\boldsymbol{\Theta^0}d\boldsymbol{\Theta^1} \tag{4}$$

A brute force search of latent labels L and latent chatting pattern Z will induce prohibitively high computational complexity. To address this challenge, we adopt the Gibbs Sampling [18] method to efficiently estimate the posterior estimates of L and Z. The Gibbs Sampling is a standard technique used to infer hidden parameters and latent variables in Bayesian Networks. The sampling procedure follows the generation process of LSCD. Finally, the inferred labels (i.e., L) are used to identify copyright-infringing videos.

IV. Evaluation On Real World Data

In this section, we evaluate StreamGuard using two real-world datasets collected from YouTube. The results demonstrate that StreamGuard significantly outperforms several representative baselines as well as the commercial solution ContentID from YouTube at the time of writing [1]. Next, we describe the datasets, experiment setup and the performance evaluation in details.

A. Datasets

We summarize the two real-world datasets (NBA and Soccer) used for evaluation in Table IV. The NBA dataset includes 53 live video streams related to the NBA basketball games with chat messages. Within the 53 video streams, 43.4% of them are copyright-infringing. The Soccer dataset contains 92 live videos with chat messages related to soccer matches in major soccer leagues worldwide. 20.65% of these soccer-related video streams are found to be copyright-infringing. We use our online crawler system (described in Section III) to collect these live videos. For each video, we started the crawling process at the beginning of the scheduled event), and collected data for a total of 30 minutes.

Table IV: Data Trace Statistics

Data Trace	NBA	Soccer
Collection Period	Dec. 2017 - March 2018	Sept. 2017 - Mar. 2018
Number of Videos	53	92
% of Copyright Infringing Videos	43.4%	20.6%
Number of Chat Users	1,635	3,149
Number of Chat Messages	57,293	87,132

To build the terminology database for extracting the indicators from the live chat messages, we collected terms related to the sporting events from sports websites such as ESPN [3], and FOX Sports [4]. We refer more details of building the terminology database in [2].

To get the ground truth label of each collected video stream, we assigned three independent labelers to manually looked through the screenshots and labeled the videos as copyright-infringing if it was streaming the actual game. Majority voting was performed to eliminate possible bias in the labeling process.

B. Baselines

We compare the following schemes to evaluate the performance of StreamGuard:

- $\mathbf{StreamGuard_{All}}$: StreamGuard system with all semantic features (i.e., both observation score and emotion score).
- $\mathbf{StreamGuard_{Obs}}$: StreamGuard system with only observation score extracted from the chat messages.
- $\mathbf{StreamGuard_{Emo}}$: StreamGuard system with only emotion score extracted from the chat messages.
- $\mathbf{Voting_{Obs}}$: A heuristic baseline that categorizes the video as copyright-fringing or not based on the average observation score extracted from the chat messages of the video (e.g., a high observation score of a video is more likely to indicate that the video is copyright-infringing).
- $\mathbf{Voting_{Emo}}$: A heuristic baseline that categorizes the video as copyright-fringing or not based on the average emotion score extracted from the chat messages of the video (e.g., a high emotion score of a video is more likely to indicate that the video is copyright-infringing).
- **BOW** : The chat messages of a video are treated as a text document. We leverage K-means ($K = 2$) [16] to cluster all "documents" (videos) into two clusters based on their TF-IDF features [19]. The videos in the cluster with a higher average observation score are classified as copyright-infringing.
- **LDA** : We first extract the topic distribution of chat messages of each video using Latent Dirichlet allocation (LDA). We then cluster the videos into two

[3]http://www.espn.com/

[4]https://www.foxsports.com/

clusters based on their topic distributions using Fuzzy K-means [20]. The videos in the cluster with a higher average observation score are classified as copyright-infringing.

- **ContentID** : YouTube's copyright infringement detection system [3].

For the baselines that contain parameter(s), we set the parameter(s) of the baselines that give them the best performance. In particular, we set the classification threshold for $Voting_{Obs}$ to decide a video if copyright-infringing as 0.2 and 0.18 for the NBA and Soccer dataset respectively. For $Voting_{Emo}$, we set the classification threshold as 0.02 for both datasets. For LDA, we set the total number of topics as 15. For the proposed StreamGuard scheme, we set the total number of latent user chatting patterns (i.e., K) as 4. The sensitivity of K is discussed in Section IV-E. We ignore the users with chat messages less than two since it gives too little information to infer their latent chatting patterns.

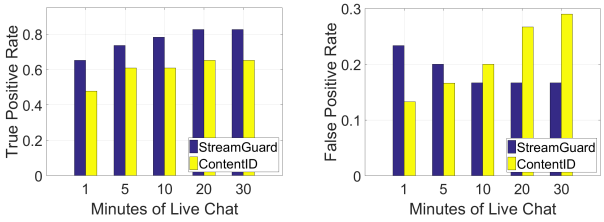
Note that we cannot directly access the actual ContentID system (because it is a *proprietary* system without an open implementation). We consider a video has been labeled by ContentID as "copyright-infringing" if it i) abruptly went offline during the broadcast, or ii) it was taken down due to the claim filed by the copyright owner. Both of these copyright-infringing cases can be verified through the screenshots we collected from YouTube.

C. Results: Detection Accuracy

We first evaluate the detection performance of StreamGuard and baselines in terms of *Accuracy*, *Precision*, *Recall* and *F1-Score*. The results are reported in Table V. We observe that StreamGuard with all semantic features achieves the best performance among all compared schemes. In particular, the StreamGuard [5] has achieved 17.02%, and 7.71% higher F1-Score compared to the best-performed baseline (i.e., ContentID from Youtube) in the NBA and Soccer datasets respectively. Voting, BOW, and LDA all have relatively poor performance in both datasets. In particular, we observe that the voting schemes often fail to distinguish between the videos that have similar observation or emotion scores. BOW and LDA schemes, on the other hand, fail to distinguish videos that have similar chat message contents. For example, users watching an NBA 2K game (non-infringing) can post similar messages as users who are watching an actual NBA match (copyright-infringing). We attribute the performance gain of StreamGuard to the explicit incorporation of latent user chatting patterns that can better identify copyright-infringing videos. We also observe that ContentID suffers from both high false positives (i.e., falsely taking down legal videos) and false negatives (i.e., missing the detection of copyright-infringing videos).

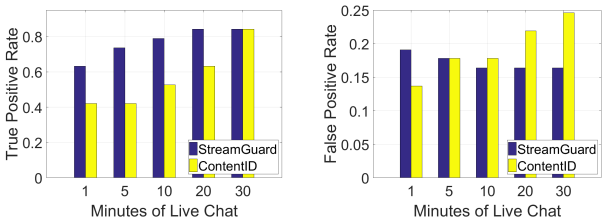
[5]In the rest of the paper, we use $StreamGuard$ to represent $StreamGuard_{All}$ when there is no ambiguity.

D. Results: Detection Time



(a) True Positive Rate (b) False Positive Rate

Figure 6: NBA Dataset



(a) True Positive Rate (b) False Positive Rate

Figure 7: Soccer Dataset

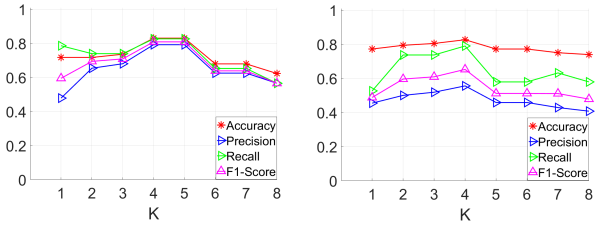
Next, we evaluate the time it takes for StreamGuard and the best-performed baseline ContentID to detect copyright-infringing videos after the video starts broadcasting. We focus on two evaluation metrics: i) the *true positive rate* that represents the ability of the scheme to correctly catch copyright-infringing videos, and ii) the *false positive rate* that characterizes the ability of the scheme to keep legal videos from being misclassified as copyright-infringing. In the experiment, we choose a set of time windows from 1 minute to 30 minutes for each video. The StreamGuard is only allowed to use the chat messages within each time window. The results are reported in Figure 6 and Figure 7. We can observe that StreamGuard has a higher true positive rate than ContentID at all time windows. The most significant performance gain is achieved at the beginning of the video stream, which suggests that StreamGuard can capture copyright-infringing videos much faster than ContentID. In terms of false positive rate, the StreamGuard incurs more false positives at the early stage of the video due to the lack of chat messages but is able to gradually catch up and eventually outperform ContentID. In contrast, ContentID mistakenly takes down more legal videos as the live broadcast goes on, which could discourage benign streamers to share their legal live videos.

E. Results: Sensitivity of Model Parameters

Finally, we evaluate the parameter sensitivity of the proposed scheme. A key parameter of StreamGuard is K - the total number of user chatting patterns in LSCD model. We

Table V: Copyright Infringement Detection Performance of All Schemes

Schemes	NBA				Soccer			
	Accuracy	Precision	Recall	F1-Score	Accuracy	Precision	Recall	F1-Score
StreamGuard$_{\mathbf{All}}$	**0.8302**	**0.7917**	**0.8261**	**0.8085**	**0.8370**	**0.5714**	**0.8421**	**0.6809**
StreamGuard$_{\mathbf{Obs}}$	0.7170	0.6333	0.8261	0.7170	0.8043	0.5217	0.6316	0.5714
StreamGuard$_{\mathbf{Emo}}$	0.6038	0.5294	0.7826	0.6316	0.6957	0.3548	0.5789	0.4400
Voting$_{\mathbf{Obs}}$	0.7170	0.7857	0.4783	0.5946	0.6739	0.3591	0.7368	0.4828
Voting$_{\mathbf{Emo}}$	0.6415	0.7500	0.2609	0.3871	0.7717	0.3750	0.1579	0.2222
BOW	0.6038	0.6250	0.2174	0.3226	0.6848	0.2917	0.3684	0.3256
LDA	0.6226	0.6000	0.3914	0.4737	0.5978	0.2955	0.6842	0.4127
ContentID	0.6792	0.6250	0.6522	0.6383	0.7717	0.4706	0.8421	0.6038



(a) NBA Dataset (b) Soccer Dataset

Figure 8: Performance w.r.t # of User Chatting Patterns

vary K from 1 to 8 and the results are shown in Figure 8. We observe that our scheme achieves the optimal performance when K is set to 4 (NBA has tied performance when K is 4 and 5). We also find the performance degrades when K becomes large due to the potential over-fitting problem.

V. Discussion and Future Work

We identified several interesting directions for future work. First, StreamGuard is primarily evaluated using YouTube videos. However, we keep the core design of StreamGuard to be generic and the only application/platform specific component is the data crawler. We plan to extend StreamGuard by developing new data crawlers for more video sharing platforms (e.g., Twitch and Vimeo) and explore the difference of user chatting patterns on these platforms. Second, StreamGuard requires the collection of keywords for several semantic indicators. In the future work, we plan to adopt the entity extraction technique [21] to automatically learn the relevant entities and keywords for each indicator. Third, it is also worth mentioning that YouTube's ContentID could have already taken down videos that never got exposed to Youtube's search engine. In this case, the videos crawled by StreamGuard could be a set of "hard cases" for ContentID. However, the evaluation results clearly suggest that StreamGuard wins ContentID over these hard cases. A promising application scenario is to use StreamGuard as an extra filter for commercial platforms such as ContendID to further filter copyright infringements. Finally, StreamGuard could invite adversarial users to degrade the performance of the system by spamming the live chats with completely random or unrelated messages. To alleviate this problem, we plan to leverage the techniques in bot/spam detection [22] and fact-checking [23], [24] to identify and depreciate the messages from potential adversarial users.

VI. Related Work

A. Video Recommendation System

The goal of online video recommendation systems is to provide personalized recommendations that help users to find videos relevant to their interests [25], [26]. For example, the recommendation system of YouTube recommends videos based on the meta-data of the video (e.g., view counts, sharing activity, upload time, etc;) and the specific interest of a user [27]. Yan *et al.* developed a video recommendation system that extracts the demographic information and preferences of users from their Twitter feeds to boost the recommendation performance [28]. Choi *et al.* proposed a novel online video recommendation system that leverages the facial expressions of users to track their dynamic preferences of videos [29]. However, the above systems do not explicitly address the copyright-infringing issue in their recommendation process. In this paper, we develop StreamGuard to address copyright infringement detection problem in online video sharing and recommendation systems.

B. Video Copy Detection

A potential solution to the copyright infringement detection of video content is called *video copy detection* [30]. The idea of this technique is to detect illegally copied videos by comparing them to the original content. For example, Thomas *et al.* developed a simple video copy detection scheme that compares videos based on the color correlation histograms extracted from the video frames [31]. Nie *et al.*

developed a video detection framework by using a tensor model to detect near-duplicate videos [8]. Hampapur *et al.* proposed a content-based video copy detection scheme that compares videos based on the global descriptions (i.e., motion and color) of the videos [32]. However, these content-based methods cannot handle the unique challenge where malicious streams can modify the video presentations to appear to be very different from the original content. In contrast, the StreamGuard system avoids directly analyzing the visual content of the videos but only relies on the chat messages from the audience.

C. Copyright Protection

The copyright protection of digital contents has become a critical undertaking for online data sharing platforms [33], [34]. Digital watermarking [35] is one of the most widely used copyright protection techniques. By using this technique, content owners can covertly embed owner information into a copyrighted material without affecting the perceived visual quality of the original content [6].Another technology for copyright protection is digital fingerprint which refers to a set of compact digital features extracted from the original content [5]. For example, Davis *et al.* developed a digital fingerprint system that generates and compares fingerprints of videos based on their pixel-level residual macro-block features [36]. However, the above copyright protection techniques focus on the *static contents* and cannot be applied to *live video streams* studied by this paper. The most relevant work is a *supervised* classifier that is recently developed to detect copyright-infringing video streams based on chat contents [2]. However, this supervised approach depends heavily on the training data and does not work for the video streams with insufficient training data. The StreamGuard complements the supervised detector by developing a principled unsupervised solution to address the problem of lack of training data for the copyright detection problem in live video streams.

VII. Conclusion

In this paper, we develop a StreamGuard system that is dedicated to detecting copyright-infringing live videos for large-scale online video sharing systems. To the best of our knowledge, StreamGuard is the first *unsupervised non-commercial* detection system that targets at finding copyright-infringing videos in live streams. StreamGuard develops a principled LSCD model that jointly estimates the latent user chatting pattern and the copyright infringement labels of the live videos. Without the requirement for training data, StreamGuard is able to handle cold-start scenarios where only a few live streams on a piece of copyrighted content are available. StreamGuard is also designed to be robust against manipulation of the visual content of the live video streams by relying only on the live chat of users. The evaluation results using two real-world live video stream datasets showed that StreamGuard can significantly outperform existing copyright detectors (i.e., ContentID from YouTube) in terms of accuracy and timeliness.

Acknowledgement

This research is supported in part by the National Science Foundation under Grant No. CNS-1831669, CBET-1637251, CNS-1566465, and IIS-1447795, Army Research Office under Grant W911NF-17-1-0409, Google 2017 Faculty Research Award. 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] M. Martemucci and A. Swerdlow, "Video game streaming brings new level of copyright issues," 2017. [Online]. Available: https://www.law360.com/articles/920036/video-game-streaming-brings-new-level-of-copyright-issues

[2] D. Y. Zhang, Q. Li, H. Tong, J. Badilla, Y. Zhang, and D. Wang, "Crowdsourcing-based copyright infringement detection in live video streams," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 367–374.

[3] D. King, "Latest content id tool for youtube," *Google Blog*, 2007.

[4] Y. Zhang, D. Zhang, N. Vance, Q. Li, and D. Wang, "A light-weight and quality-aware online adaptive sampling approach for streaming social sensing in cloud computing," in *icpads*. IEEE, 2018.

[5] A. Barg, G. R. Blakley, and G. A. Kabatiansky, "Digital fingerprinting codes: Problem statements, constructions, identification of traitors," *IEEE Transactions on Information Theory*, vol. 49, no. 4, 2003.

[6] C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE Journal on selected areas in communications*, vol. 16, no. 4, pp. 525–539, 1998.

[7] M. M. Esmaeili, M. Fatourechi, and R. K. Ward, "A robust and fast video copy detection system using content-based fingerprinting," *IEEE Transactions on information forensics and security*, vol. 6, no. 1, 2011.

[8] X. Nie, Y. Yin, J. Sun, J. Liu, and C. Cui, "Comprehensive feature-based robust video fingerprinting using tensor model," *IEEE Transactions on Multimedia*, vol. 19, no. 4, pp. 785–796, 2017.

[9] D. Wang, B. K. Szymanski, T. Abdelzaher, H. Ji, and L. Kaplan, "The age of social sensing," *arXiv preprint arXiv:1801.09116*, 2018.

[10] Y. Zhang, N. Vance, D. Zhang, and D. Wang, "On opinion characterization in social sensing: A multi-view subspace learning approach," in *2018 14th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2018, pp. 155–162.

[11] 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 *Big Data (Big Data), 2018 IEEE International Conference on*. IEEE, 2018.

[12] D. Wang, M. T. Amin, S. Li, T. Abdelzaher, L. Kaplan, S. Gu, C. Pan, H. Liu, C. C. Aggarwal, R. Ganti, X. Wang, P. Mohapatra, B. Szymanski, and H. Le, "Using humans as sensors: An estimation-theoretic perspective," in *Proc. 13th Int Information Processing in Sensor Networks Symp. IPSN-14*, Apr. 2014, pp. 35–46.

[13] D. Wang, T. Abdelzaher, L. Kaplan, and C. C. Aggarwal, "Recursive fact-finding: A streaming approach to truth estimation in crowdsourcing applications," in *The 33rd International Conference on Distributed Computing Systems (ICDCS'13)*, July 2013.

[14] N. J. Gadgil, K. Tahboub, D. Kirsh, and E. J. Delp, "A web-based video annotation system for crowdsourcing surveillance videos," in *Imaging and Multimedia Analytics in a Web and Mobile World 2014*, vol. 9027. International Society for Optics and Photonics, 2014, p. 90270A.

[15] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, E. Dempsey *et al.*, "Textblob: simplified text processing," *Secondary TextBlob: Simplified Text Processing*, 2014.

[16] S. C. Dharmadhikari, M. Ingle, and P. Kulkarni, "Empirical studies on machine learning based text classification algorithms," *Advanced Computing*, vol. 2, no. 6, p. 161, 2011.

[17] D. M. Blei, "Probabilistic topic models," *Communications of the ACM*, vol. 55, no. 4, pp. 77–84, 2012.

[18] C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan, "An introduction to mcmc for machine learning," *Machine learning*, vol. 50, no. 1-2, pp. 5–43, 2003.

[19] R. C. Balabantaray, C. Sarma, and M. Jha, "Document clustering using k-means and k-medoids," *arXiv preprint arXiv:1502.07938*, 2015.

[20] C.-H. Li, B.-C. Kuo, and C.-T. Lin, "Lda-based clustering algorithm and its application to an unsupervised feature extraction," *IEEE Transactions on Fuzzy Systems*, vol. 19, no. 1, pp. 152–163, 2011.

[21] O. Etzioni, M. Cafarella, D. Downey, A.-M. Popescu, T. Shaked, S. Soderland, D. S. Weld, and A. Yates, "Unsupervised named-entity extraction from the web: An experimental study," *Artificial intelligence*, vol. 165, no. 1, pp. 91–134, 2005.

[22] A. H. Wang, "Detecting spam bots in online social networking sites: a machine learning approach," in *IFIP Annual Conference on Data and Applications Security and Privacy*. Springer, 2010, pp. 335–342.

[23] D. Wang, L. Kaplan, H. Le, and T. Abdelzaher, "On truth discovery in social sensing: A maximum likelihood estimation approach," in *Proc. ACM/IEEE 11th Int Information Processing in Sensor Networks (IPSN) Conf*, Apr. 2012, pp. 233–244.

[24] D. Y. Zhang, J. Badilla, Y. Zhang, and D. Wang, "Towards reliable missing truth discovery in online social media sensing applications," in *2018 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*. IEEE, 2018, pp. 143–150.

[25] C. A. Gomez-Uribe and N. Hunt, "The netflix recommender system: Algorithms, business value, and innovation," *ACM Transactions on Management Information Systems (TMIS)*, vol. 6, no. 4, p. 13, 2016.

[26] K. Li, S. Li, S. Oh, and Y. Fu, "Videography-based unconstrained video analysis," *IEEE Transactions on Image Processing*, vol. 26, no. 5, pp. 2261–2273, 2017.

[27] J. Davidson, B. Liebald, J. Liu, P. Nandy, T. Van Vleet, U. Gargi, S. Gupta, Y. He, M. Lambert, B. Livingston *et al.*, "The youtube video recommendation system," in *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 2010, pp. 293–296.

[28] M. Yan, J. Sang, and C. Xu, "Unified youtube video recommendation via cross-network collaboration," in *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 2015, pp. 19–26.

[29] I. Y. Choi, M. G. Oh, J. K. Kim, and Y. U. Ryu, "Collaborative filtering with facial expressions for online video recommendation," *International Journal of Information Management*, vol. 36, no. 3, pp. 397–402, 2016.

[30] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *Proceedings of the 6th ACM international conference on Image and video retrieval*. ACM, 2007, pp. 371–378.

[31] R. M. Thomas and M. Sumesh, "A simple and robust colour based video copy detection on summarized videos," *Procedia Computer Science*, vol. 46, pp. 1668–1675, 2015.

[32] A. Hampapur, K. Hyun, and R. M. Bolle, "Comparison of sequence matching techniques for video copy detection," in *Storage and Retrieval for Media Databases 2002*, vol. 4676. International Society for Optics and Photonics, 2001, pp. 194–202.

[33] T.-Y. Chung, M.-S. Hong, Y.-N. Oh, D.-H. Shin, and S.-H. Park, "Digital watermarking for copyright protection of mpeg2 compressed video," *IEEE Transactions on Consumer Electronics*, vol. 44, no. 3, pp. 895–901, 1998.

[34] M. K. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and applications of digital watermarking and content protection*. Artech House, 2003.

[35] C.-Y. Lin, "Watermarking and digital signature techniques for multimedia authentication and copyright protection," Ph.D. dissertation, Columbia University, 2001.

[36] A. G. Davis, "Digital video fingerprinting," May 3 2016, uS Patent 9,330,426.