# Scalable Misbehavior Detection in Online Video Chat Services

Xinyu Xing [1], Yu-li Liang [2], Sui Huang [3], Hanqiang Cheng [4],
Richard Han [2], Qin Lv [2], Xue Liu [4], Shivakant Mishra [2], Yi Zhu [5]
[1]Georgia Institute of Technology, [2]University of Colorado - Boulder,
[3]Ohio State University, [4]McGill University, [5]Tsinghua University
xxing8@gatech.edu, {richard.han, qin.lv}@colorado.edu, xueliu@cs.mcgill.ca

## ABSTRACT

The need for highly scalable and accurate detection and filtering of misbehaving users and obscene content in online video chat services has grown as the popularity of these services has exploded in popularity. This is a challenging problem because processing large amounts of video is compute intensive, decisions about whether a user is misbehaving or not must be made online and quickly, and moreover these video chats are characterized by low quality video, poorly lit scenes, diversity of users and their behaviors, diversity of the content, and typically short sessions. This paper presents EMeralD, a highly scalable system for accurately detecting and filtering misbehaving users in online video chat applications. EMeralD substantially improves upon the state-of-the-art filtering mechanisms by achieving much lower computational cost and higher accuracy. We demonstrate EMeralD's improvement via experimental evaluations on real-world data sets obtained from Chatroulette.com.

## Categories and Subject Descriptors

K.4.1 [**Computers and Society**]: Public Policy Issues–abuse and crime involving computers; K.4.2 [**Computers and Society**]: Public Policy Issues–human safety, abuse and crime involving computers

## Keywords

Online video chat, misbehavior detection, video safety

## 1. INTRODUCTION

The popularity of online video chat services has been increasing over the last few years. Web services such as Chatroulette [1], myYearbook [2], Omegle [3] and TinyChat [4] have all been experiencing aggressive membership growth. For example, Chatroulette had more than 20 million visitors per month by May 2011 [5], which is three times the number of visitors in July 2010 [6]. The common feature of an online video chat website is that it *randomly* pairs online users from around the world for webcam-based conversations. These users can then conduct online chat via video, audio and text with one another. At any point, each user may leave the current chat and seek another random user for chatting. In general, such websites are offered for free and are easy to use, which enhances their popularity.

However, a critical problem encountered by these online video websites is that they attract a large number of misbehaving users who expose themselves, and/or broadcast offensive, obscene or pornographic content. For instance, our observation on a typical weekend (summer 2011) from a representative online video chat website (Omegle) indicates that 35% of the videos broadcast by this website have nudity in them. This is a major problem since a large fraction of the online video chat users are underage minors – about 1/4th by our estimates – and are thus exposed, perhaps illegally, to content unsuitable for their age.

Researchers have recently begun to address this problem of detecting and filtering misbehaving users and inappropriate content in online video chat systems. The SafeVchat system [7] employs a fusion technique that integrates the results from multiple image-based classifiers to develop a stronger inference about whether a particular video chat user is misbehaving or not. This fusion approach overcomes several major challenges in detecting and filtering misbehaving users in online video chat systems, including low quality Webcam video, poorly lit scenes, and diversity of users and their behaviors. Starting in early 2011, this system was successfully deployed on Chatroulette.com, the leading random video chat site on the Web, and has helped reduce the percentage of misbehaving users from about 30% down to about 2-5% today.

However, there are a number of key limitations with prior work. Experiences with deployment in Chatroulette.com have shown that SafeVchat requires over a hundred servers working at near full CPU utilization 24 hours a day to handle the Chatroulette user load. Chatroulette has found this to be an expensive solution that is also not very scalable. Furthermore, Chatroulette has found that SafeVchat's accuracy was only acceptable enough to identify normal users with high accuracy, not misbehaving users. Given that most users are normal users, Chatroulette employs a two-stage solution: SafeVchat is first used to identify and filter out most normal users; in the second stage, a large number of online human users are employed to review any remaining sessions, which could have either misbehaving users or normal ones missed by the first stage. The large bank of human reviewers introduces a second cost factor, and is also not scalable, especially for many of the smaller companies emerging in this application domain. Only well-funded Web companies like Chatroulette can afford such a server-intensive and human-intensive solution. As a result of these costs, another leading video chat site, Omegle chose not to deploy this solution.

In this paper, we present a new approach to misbehavior detection in online video chat systems that significantly improves upon the state of the art in terms of its increased scalability while also


*KDD'12,* August 12–16, 2012, Beijing, China.

achieving higher accuracy. We term our approach EMeralD (Efficient Misbehavior Detection). The key problem that EMeralD solves is how to improve scalability while preserving or even improving accuracy. It is easy to improve scalability of misbehavior detection by simply performing less classification, but this reduces accuracy. The challenge is to improve scalability while simultaneously minimizing the impact on accuracy. We accomplish this by first observing that the reason SafeVchat is not very scalable is that it executes every classifier on every image snapshot before performing its fusion algorithm. Given N classifiers, it fails to note which of the N classifiers causes the overall decision to exceed the probability threshold for making a decision. As a result, some classifiers are executed that do not need to be, and these may be computationally intensive. For example, a face classifier may be sufficient to prove the existence of a normal user, while additional information from the skin classifier may be unnecessary.

EMeralD implements a more efficient approach by measuring at each step whether the execution of a particular classifier (in reality, EMeralD uses rules that aggregate classifiers) has put the overall decision probability over the acceptable threshold for identification, in our case for normal users. In this way, we are able to execute fewer rules/classifiers and thus improve the computational efficiency and scalability while achieving the same probability threshold of success that was acceptable for SafeVchat. A difficult problem that EMeralD solves is what is the optimal ordering of rule execution that minimizes the computational latency? What makes this problem challenging is that each rule has a different coverage, namely the number of normal users with features who could be filtered out by that rule. Thus, the number of users remaining after the previous stage of rule filtering changes as the permutation of rules is changed. EMeralD proposes to use an A*-search algorithm to determine the optimal ordering of the rules to minimize computational latency by exploiting coverage relationships.

However, we noticed that there was a further opportunity to improve accuracy even as we were improving scalability. First, we observed that a key missed opportunity with prior work was that each classifier was viewed as independent from others, and thus failed to exploit semantic correlations between the classification results. EMerald instead exploits these correlations, such as spatial correlations and size correlations between different image features. For example, we know that a mouth should reside within a face. Therefore, if a face classifier finds a face in an image, and a mouth classifier detects a mouth in the same image, then EMerald makes a stronger inference that there is indeed a human face in the image if the mouth's location is within the face's location. Such semantic correlations are exploited by EMeralD to improve the accuracy of classification while using the same OpenCV classifiers as SafeVchat.

EMeralD has been extensively evaluated in terms of its accuracy (precision/recall) and computational cost on real-world datasets obtained from Chatroulette. This evaluation demonstrates that EMeralD outperforms by a large margin the current state-of-the-art detection and filtering techniques such as PicBlock and Bag-of-Visual-Words-based detection. Furthermore, EMeralD is shown to be much more computationally efficient (31%-79% reduction in per-user latency) and scalable than SafeVchat, while also achieving greater accuracy than SafeVchat. For example, to process 11,000,000 image snapshots per hour, Chatroulette requires 182 servers with full CPU usage to detect misbehaving users with SafeVchat, while only 57 servers are needed with EMeralD. By improving both scalability and accuracy, EMeralD makes misbehaving user detection software much more practical and affordable to a wider range of online video chat services. EMeralD is currently being evaluated by three industrial companies at present under an evaluation license, including Chatroulette. In general, EMeralD's accuracy and scalability makes it suitable for most video-based, realtime interactive applications on the Internet.

## 2. RELATED WORK

Techniques to detect pornographic content can be divided into three categories: manual crowd-sourcing, skin color based detection, and Bag-of-Visual-Words (BoVW) based detection. Manual crowd-sourcing consists of human reviewers inspecting video snapshots. YouTube [8], for example, allows users to flag and report inappropriate content presented on their website [9], which are then reviewed by moderation teams of YouTube. Crowd-sourcing has also been used in online video chat websites such as Chatroulette [1], myYearbook [2] and Tinychat [4]. However, this approach incurs high economic cost and thus is not scalable, is not applied uniformly, i.e., only images that are "reported" are actually inspected, does not report all misbehaving users, and falsely reports some normal users due to pranksters. Online video chat services have stopped using this mechanism for these limitations.

Skin color based detection mechanisms are broadly used and achieve acceptable performance in terms of precision and recall for pornographic content detection [10][11][12][13][14]. This approach identifies skin exposure regions in an image using a statistical color model. Size, texture and shape of the skin exposure regions are sometimes considered to further improve performance.

While skin color based detection mechanism has proved to be effective and efficient in the context of pornographic image detection, its effectiveness is limited in the context of online video chat systems. Due to the diverse quality of snapshot images captured from online video chat systems, the statistical color model is insufficient for identifying misbehaving users. While pornographic images are usually taken by professional cameras under good lighting conditions, the images in online video chat services are taken by chatters' poor-quality webcams, which significantly affects the appearance of the skin. In addition, since different users may be under fairly diverse illumination conditions, skin color in snapshot images have significant variance. Indeed, a recent survey [15] concludes that skin color based detection mechanism may only be used as a preprocessor for pornographic content detection, and other content types such as textual content [12], motion analysis [16] and structural content [11] need to be incorporated to improve accuracy.

There are two recently-proposed systems that harness the Bag of Visual Words model (BoVW framework) to detect pornographic images [17][18]. In a BoVW framework, the Scale-Invariant Feature Transform (SIFT) [19] extracts feature descriptors of an image. Experimental results over data sets containing commercial pornographic content, shown in [17][18], demonstrate a significant performance improvement in terms of precision and recall. However, our experiments (described in Section 6) indicate that a BoVW-based detection mechanism performs poorly in the context of online video chat systems. First, SIFT descriptors are keypoint descriptors that are good at describing salient regions. However, in images that are taken under dark illumination conditions (a relatively common condition in online video chat systems), only a few salient keypoints can be found. Second, the problem we confront here is more difficult than the pornography detection problem due to the smaller inter-class distance between different categories. For example, the "difference", or visual distance, between the fully clothed category and the nude body trunk category is large. However, for our problem, the difference between misbehaving users and normal users is not that clear. Both normal and misbehaving users can be partially clothed (normal male users show partially

naked upper body while misbehaving ones their genitals partially clothed). Third, the BoVW based detection mechanism is compute intensive. We implemented "BoVW + HueSIFT" approach proposed in [18] using the implementation of HueSIFT proposed in [20], and found that it takes 1960 milliseconds to classify one image.

To address the drawbacks of these detection mechanisms, we proposed SafeVchat [21]. SafeVchat harnesses Dempster-Shafer Theory to calculate the probability that a user is misbehaving. Though SafeVchat provides acceptable classification performance in terms of precision and recall, it suffers from two key limitations. First, it incurs over 1200 milliseconds of computational latency for each user, which results in large computational resource requirements. Second, it results in 3% leakage of detecting misbehaving users. A detailed analysis is provided in Section 6.

# 3. SYSTEM OVERVIEW

## 3.1 Design Requirements

Our obscene content detection system should satisfy two key requirements - scalability and precise classification. Based on our experiences, online video chat systems are capable of providing periodic image snapshots of users (it is once every 30 seconds for Chatroulette). However, image processing is quite computationally intensive, and as mentioned earlier requires a large array of servers all working near full CPU utilization. Therefore, our first design objective is to limit EMeralD's consumption of computational resources. This is achieved by intelligently choosing which filters to activate and in what order.

Second, our goal is to achieve high precision and recall in terms of correctly classifying misbehaving users. The precision should be high because in a system such as Chatroulette, all users classified as misbehaving will be subject to manual review by human monitors. If too many normal users are incorrectly classified as misbehaving, then the burden of these false positives on the human monitors will be high, incurring a high labor cost. Recall should also be high, i.e., most misbehaving users should be detected and only limited few may be falsely classified as normal and appear in the chat system.

Our design leverages our observations of users of the online video chat systems Chatroulette, Omegle, and myYearbook. Misbehaving users on online video chat systems usually hide their faces during the conversation. Some misbehaving users do not completely expose themselves, e.g. expose only their genitals in front of the webcam and stay partially clothed. Chatters who present their faces in front of webcams are mostly normal users because a majority of webcams only provide a narrow field of view (i.e., no wide angle lens are installed onto webcams); thus showing both the body trunk and the face of a user requires the user placing his/her webcam far from the user. However, chatters who do not show their faces may not be flashers. A fair amount of chatters do not show their faces clearly, i.e. only a partial face is presented in front of the webcam. Webcams are usually set up in a fixed position and are not moved often.

## 3.2 System Architecture

The key contribution of EMeralD is to offer a system that can optimize for latency/scalability while enhancing accuracy. Earlier systems had to execute every individual classifier before a fusion decision - misbehaving or not - could be made [7]. Having to execute every classifier incurred a large computational expense. Instead, EMeralD orders its filtering rules in an optimal sequence that minimizes the overall latency incurred during classification. In our new approach, not all individual classifiers need be executed in order to arrive at an overall classification decision.

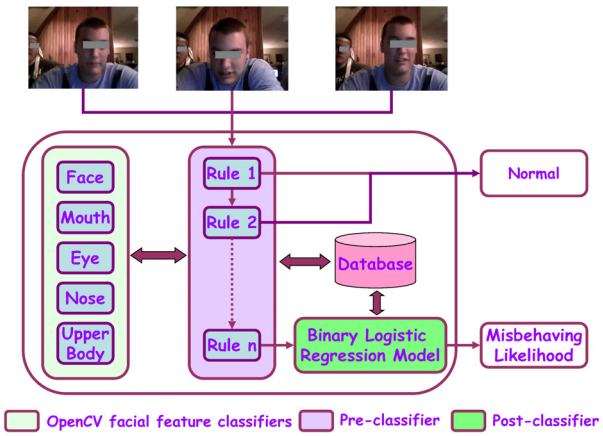



**Figure 1: EMeralD: System architecture for detecting misbehaving users in online video chat services.**

As shown in Figure 1, we partition the system into a rule-based pre-classifier or front end and a binary logistic regression model or back end. The rule-based pre-classifier leverages data obtained from five OpenCV facial feature classifiers (face, mouth, eye, nose, upper body). Three sequential snapshot images from a user are first delivered to the pre-classifier. In the pre-classifier, a set of association rules are defined in advance and the user's three snapshot images are sequentially examined by pre-defined rules. During the examination of the rules, the facial feature classifiers of OpenCV are called. The pre-classifier stores the classification output from each of the facial feature classifiers into a centralized database.

The rule checking in the pre-classifier is performed following a specific sequence, so that when the user's three snapshot images satisfy a specific rule, the user will be immediately classified as a normal user and the rest of the rule checking operations will not be performed. Therefore, the rule examination in the pre-classifier can filter out a significant number of normal users whose snapshot images do not need to pass all the operations of the facial feature classifiers. Notice that each rule in the pre-classifier needs to use the outputs of partial facial feature classifiers. To circumvent redundant computation from the facial feature classifiers of OpenCV, we cache the outputs of facial feature classifiers in a centralized database, so they can be reused by other classifiers.

In EMeralD, if there are no rules matching with the user's snapshots, then the binary logistic regression of the post-classifier is invoked. After obtaining the user's three snapshot images, the post-classifier will first retrieve the outputs of the facial feature classifiers for these three snapshots from the centralized database. It then calls a motion-based skin color detector [21] to obtain the skin exposure proportions of the user. By combining both the skin exposure proportions of a user and the user's facial features, the post-classifier harnesses a binary logistic regression model to predict the probability of being a misbehaving user for the user. Binary logistic regression is adept at identifying misbehaving users, but is costly to compute, and thus we do not invoke this operation until a large fraction of normal users have already been filtered out.

# 4. RULE-BASED PRE-CLASSIFIER

In order to improve scalability, we take advantage of early classification results, thereby exiting the decision-making process as soon as an identification threshold is reached, rather than executing

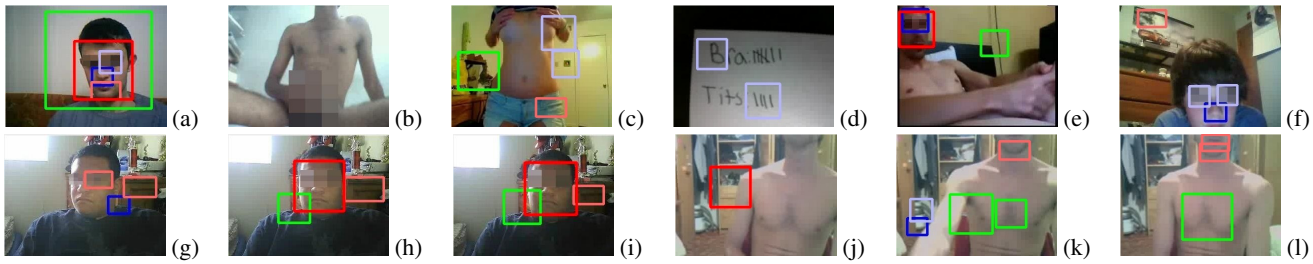



**Figure 2: Facial feature classification results of OpenCV for snapshot images from Chatroulette – upper body (green square), face (red square), mouth (pink square), nose (blue square) and eye (purple square).**

every single classifier, as in prior systems. We wish to avoid executing any more classifiers than we are required to execute. To do this, we first identify important higher-level discriminative features. We then extract rules for the most frequently occurring joint patterns among these discriminative features, using a set of representative training data. Finally, we consider all rules that achieve greater than the threshold accuracy for correctly identifying normal users, and reorder them so as to minimize overall computational latency. If any such rule identifies a user as normal, then we know that we can trust this result to be at least accurate above the threshold, and can therefore terminate the classification process early, without executing any subsequent rules.

## 4.1 Identifying Discriminative Characteristics

We need to first develop higher-level discriminative characteristics that exploit the semantic knowledge of facial features in order to improve the accuracy of EMeralD over prior work. Using the raw OpenCV classifiers, or rules based just on these classifiers, without exploiting semantic knowledge, misses the opportunity to improve the accuracy of misbehavior detection. One reason that limits the accuracy of such non-semantic approaches is that the OpenCV classifiers themselves can be inaccurate when taken in isolation, as shown in Figure 2. Thus, rules based on non-semantic approaches will also be limited in accuracy. For example, the eye classifier of OpenCV misclassifies a black character written on white paper as an eye (see Figure 2(d)). Thus, we extract higher-level discriminative characteristics from the basic outputs of facial feature classifiers of OpenCV.

Since filtering misbehaving users in the video chat context is a novel research area, there is a paucity in prior work that could guide us about which features we should consider to classify misbehaving users. Our extensive observations help us identify a long list of initial correlates to the user being a normal one or not. However, we do not make our decision simply by this heuristic examination. We relied on the statistical test results of the relationship. Only the correlates that show statistically significant relationship with the user's identity are included into our association rules. We summarize our new discriminative characteristics as follows.

(1) The presence of a face in a non-facial region is usually an occurrence by chance. As shown in Figure 2(g)-(l), mis-identified faces in non-facial regions usually only appear in one snapshot image of a user, while true faces can be correctly identified in multiple snapshot images of a user. Therefore, we define a discriminative characteristic $Face[n]$, where $n \in \{0, 1, 2, 3\}$ is the number of snapshot images of a user with at least one identified face by OpenCV. For example, the user whose snapshot images are shown in Figure 2(g)-(i) has $Face[2]$.

(2) The face classifier of OpenCV may identify multiple faces in a snapshot, and there is an extremely low random chance that the face classifier mistakenly identifies two or more faces in non-facial regions of one snapshot image. Therefore, we consider whether multiple faces appear in at least one snapshot image of a user, denoted as $MultiFace[n], n \in \{Yes, No\}$.

(3) Although a user has 0.99 likelihood of being normal when a face is present in his or her snapshot images, there are still a small number of misbehaving users who present their faces in front of the webcam. Since webcams have a narrow view angle, to show both their face and genitals, a misbehaving user has to stay far away from the webcam. Our analysis shows that faces of this type of misbehaving users are usually placed on the corner of their snapshots. Therefore, we also consider the positions of the user's faces in the user's three snapshots as a discriminative characteristic. To describe face position efficiently, we first calculate the centroid coordinate of a face. We then calculate the distance from the centroid coordinate to the bottom-left (and bottom-right) corner of the snapshot image, called "left distance" (and "right distance"). The larger of these two distances is selected as the face distance of the snapshot. Since the length of an upper body is at least two times longer than that of a face, it is difficult for a user to show both his genital and face when his face has a large face distance. We divide the face distance by the length of the face in the snapshot and use the quotient as the position of the face in the snapshot. Note that for a snapshot image with zero or multiple faces, the position of the face is defined as $\infty$ and 0. Face positions can be ranked and we select the maximum face position to represent the face position of the user within user's three snapshot images. The face position of a user is a continuous variable. To use this discriminative characteristic for association rules, we quantize the continuous values into four categories (bins), denoted as $FacePos[n], n \in \{B_1, B_2, B_3, B_4\}$.

(4) Further observations on the output of the upper body classifier of OpenCV indicate that an upper body detected in a snapshot is usually wrong, especially when the detected upper body is fairly small (see Figure 2(c)). In contrast, when the upper body classifier identifies a large region in a snapshot as an upper body, the labeled region is typically correct (see Figure 2(a)). To use the presence of an upper body as a discriminative characteristic, we consider the largest OpenCV-labeled upper body size of a user within the user's three snapshots and denote this characteristic as $UpperBody[n], n \in \{B_0, B_1, B_2, B_3, B_4\}$. Similar to the face position characteristic, we also quantize continuous upper-body size values into 5 bins and make it a categorical variable.

(5) Although the eye's visual characteristic – a center-surround pattern – is common in non-facial regions, the presence of OpenCV-labeled double eyes in a correct relative position generally represent true eyes of a user (see Figure 2(f) versus 2(d)). We consider how many snapshot images of a user contain OpenCV-labeled double eyes in a correct relative position. We denote this discriminative characteristic as $DoubleEye[n], n \in \{0, 1, 2, 3\}$ where $n$ is the number of snapshot images of a user with OpenCV-labeled double eyes in a correct relative position.

(6) Similar to eye's visual characteristic, nose, mouth, face and upper body's visual characteristics are also commonly present in

non-facial regions (see Figure 2(c), 2(j) and 2(k)). One other approach to reduce the mislabeling of the OpenCV facial feature classifiers is to combine the outputs of two facial feature classifiers and examine whether the outputs of two classifiers are in a correct relative position. A correct relative position for two different facial characteristics should satisfy at least one of the position relationships (e.g., Figure 2(a), 2(f)). Specifically, we consider 6 pairs of facial feature relationships as discriminative characteristics: a nose in a face, an eye in a face, a mouth in a face, a face in an upper body, an eye on upper left/right of a nose, and a nose on top of a mouth. Since these combinations may still appear by chance, we count the combination in user's three snapshot images and denote these combined discriminative characteristics as $NoseFace[n]$, $EyeFace[n]$, $MouthFace[n]$, $EyeNose[n]$, $FaceUpperBody[n]$, $NoseMouth[n]$, $n \in \{0, 1, 2, 3\}$.

## 4.2 Creating Association Rules

Based on the discriminative characteristics, we harness the Apriori algorithm [22] to create association rules which are used in the pre-classifier of EMeralD. We first describe each user as an 11-element vector and each element represents a discriminative characteristic. For example, the user whose snapshot images are shown in Figure 2(g)∼2(i) is described as $\{Face[2], MultiFace[No], FacePos[B_1], UpperBody[B_0], DoubleEye[0], NoseFace[0], EyeFace[0], MouthFace[0], FaceUpperBody[0], EyeNose[0], NoseMouth[0]\}$. The Apriori algorithm attempts to find all frequent $k$-itemsets ($k = 1, \ldots, 12$) and uses the corresponding frequent itemsets to create association rules. Note that there are frequent 12-itemsets because we investigate the relationship between 11 discriminative characteristics and the hypothesis of being a normal user. For a frequent $k$-itemset, the higher the value of $k$, the more computation cost an association rule will involve, because the examination of an association rule needs to involve the operations of facial feature classifiers of OpenCV which are usually compute-intensive. Rule $Face[2] \Longrightarrow User[Normal]$, for example, only involves the operation of the face classifier, while rule $Face[2]\&EyeNose[1] \Longrightarrow User[Normal]$ involves the computation of face, nose and eye classifiers. To involve as little computation and filter as many users as possible, our association rule generation and selection, in practice, have to satisfy the following conditions:

• Since we observed not all the normal users present their faces and the facial feature classifier of OpenCV may not be able to identify their facial characteristics even when they present their faces in front of webcams, we empirically set the minimum support value to 100 out of 10,000 (1%).

• An online video chat system like Chatroulette and myYearbook usually skips the costly human review process for users with high likelihood of being normal; therefore, the minimum confidence for generating association rules is 0.99.

• The association rules that are used in the pre-classifier of EMeralD have to follow the form $X_1\&X_2\&...\&X_i \Longrightarrow User[Normal]$ where $X$ denotes a discriminative characteristic and $User[Normal]$ represents the hypothesis that *a user is normal*.

• To reduce computation cost, we limit the operations of some facial feature classifiers of OpenCV for each association rule. For each association rule that is selected to use in the pre-classifier of EMeralD, we designate the number of facial feature classifiers involved by the rule to be no greater than two.

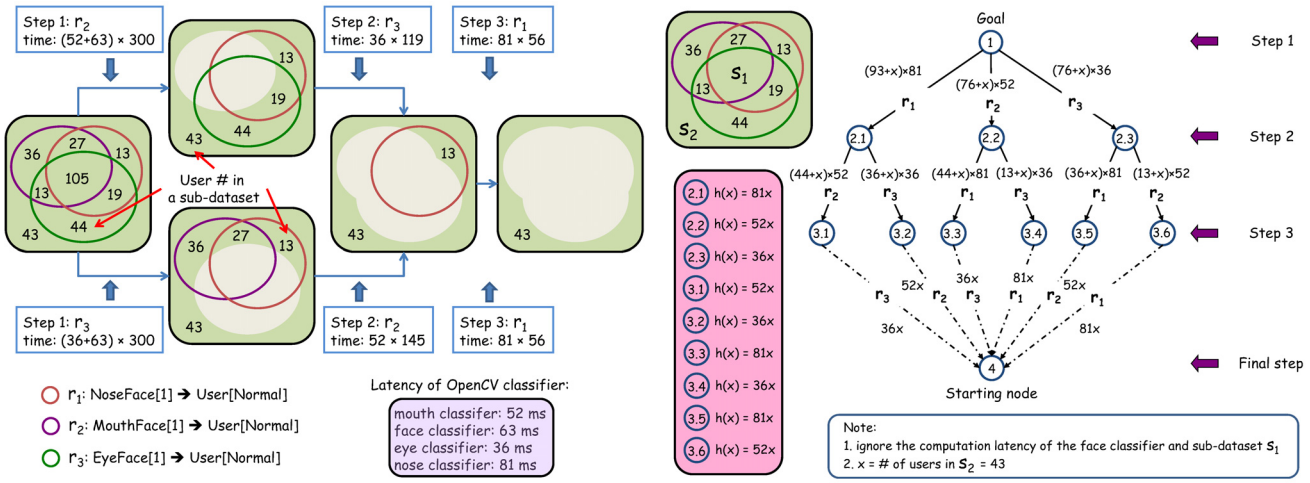
• We also constrain that the association rules used in the pre-classifier cannot be redundant. An association-rule set is non-redundant when the following is true. If there is association rule $r'$ which is generated from frequent itemsets $\Omega'$ and has confidence value $c'$, then there is no other association rule $r$ which is generated from frequent itemsets $\Omega$ ($\Omega' \subset \Omega$) and has confidence value $c$ ($c' > c$). Here, if rule $r$ is redundant, the dataset that rule $r$ covers is also covered by rule $r'$.

## 4.3 Minimizing Computational Latency

After creating the association rules, we need to further determine the operation sequence of all the association rules, i.e., in what order should the rules be examined. Each association rule has its computation cost and support (i.e., the users covered by the association rule), and a user might be covered by several different association rules because of OpenCV-labeled multiple facial features. In addition, each facial feature classifier that is used by multiple rules only need to be computed once, thus saving the computation cost. Therefore, the computation cost is significantly dependent upon the *order of rule execution*. For example, in Figure 3(a), assume that three association rules $r_1$, $r_2$ and $r_3$ can filter 257 users from a 300-Chatroulette-user dataset and label them as normal users. Each of the association rules has its own computation cost. We consider two examination sequences for these three association rules. The first examination sequence follows $r_2$, $r_3$ and $r_1$, which takes 69.60 seconds to filter out 257 users. In another sequence - $r_3$, $r_2$, $r_1$, the entire processing time for filtering out the same number of users is 41.78 seconds, which is 40% less than that of the first sequence.

To determine the order of rule execution that minimizes the computational latency, we model the problem as a path finding and graph traversal problem and apply the A* search algorithm [23] to find the least-cost path. First, we define two sub-datasets, $s_1$ and $s_2$. These sub-datasets contain those user datasets that do not affect our computation latency no matter what sequence the association rules are examined in. The sub-dataset $s_1$ is the sub-dataset which is covered by all the association rules. No matter which association rule is examined first, sub-dataset $s_1$ will be filtered. Therefore, this sub-dataset does not involve extra examination cost for other association rules irrespective of which association rule in the pre-classifier is examined first. We ignore this sub-dataset in our modeling. In Figure 3(a), there are 105 such users. The second sub-dataset - $s_2$ - is the sub-dataset which is not covered by any association rule in the pre-classifier. All datasets in this sub-dataset will be examined by all association rules, which involves the maximum computation cost (i.e. all facial feature classifiers of OpenCV will be performed for examining sub-dataset $s_2$). In Figure 3(a), there are 43 such users. In addition to sub-dataset $s_1$, our model also ignores the computation cost of the classifier which all the association rules involve. For example, Figure 3(a) indicates all three association rules involve the operations of the face classifier. No matter which association rule is examined first, the face classifier has to be used for labeling the face regions for all the 300-Chatroulette users.

Since any association rule can only cover a part of a dataset, selection of an association rule involves extra computation cost, which is the cost of applying that rule on all users not filtered out by it. For example, association rule $r_1$ does not cover $93+43$ users. So, the extra commutation cost of selecting $r_1$ is $(93+43) \times 81ms$. Notice that we have not considered the cost of face classifier, as face classifier is included in all association rules. Similarly, extra computation cost of selecting rule $r_1$ followed by rule $r_2$ is the extra computation cost of selecting $r_1$ plus $(44+43) \times 52ms$, where $44+43$ is the number of users not covered by $r_1$ or $r_2$. Finally, the extra computation cost of selecting the sequence $r_1, r_2, r_3$ is the extra computation cost of selecting $r_1$ followed by $r_2$ plus $43 \times 36ms$, where 43 is the number of users not covered by any of the three rules, i.e. sub-dataset $s_2$.




(a) Two examination sequences of three association rules.

(b) Modeling the least computation cost for three association rules.

**Figure 3: Understanding and modeling the computation cost of three association rules.**

We use the extra computation cost of each association rule and association-rule-selection steps to model a graph. Each edge in this graph represents a selection of an association rule. The weight on an edge is the extra communication cost of selecting this rule in the sequence starting from the root. In general, weight of an edge with rule $r$ is dependent on the number of users ($K$) that have not yet been filtered out and are not covered by rule $r$ and the cost ($\delta t$) of applying $r$, i.e. the edge weight is ($w = K \cdot \delta t$). Note that the weight of the edge for the same association rule is different because the value of $K$ and $\delta t$ may vary in each association-rule-selection step. For example, Figure 3(a) demonstrates that association rule $r_2$ which is examined in two different sequences has different value of $K$ ($K = 119$ and $K = 56$) and same value of $\delta t$ [1]. The connection of the nodes in the graph follows the steps of selecting the association rules (i.e., in any path of the graph from the first step to the last, each association rule can only appear once; further, a path from the first step to the last has to contain all the association rules that are used in the pre-classifier.). Figure 3(b) shows a graph which represents the example case shown in Figure 3(a).

Since A* search algorithm uses a distance-plus-cost heuristic function that contains an admissible heuristic function $h(x)$, we further define the admissible heuristic function $h(x) = \delta t \cdot x$ where $x$ is the total number of users in sub-dataset $s_2$. It is obvious that function $h(x)$ is not an admissible heuristic if we take the step-one node as the starting node and the final-step node as the goal. To ensure function $h(x)$ does not overestimate the distance to the goal, we designate the final-step and step-one nodes as the starting node and the goal, respectively.

## 5. PROBABILISTIC POST-CLASSIFIER

In stage two, all remaining users not filtered out as normal by the pre-classifier are subject to the probabilistic post-classifier, in order to identify misbehaving users.

To identify the appropriate statistical model, we need to understand the distributional characteristics of variables of interest. Since our goal is to establish a model to identify flashers, a user being a flasher or not is our dependent variable. It is a binary response with 2 categories (0, 1) and thus follows a binomial distribution. A simple linear regression model, which assumes normal distribution of dependent variable, is not appropriate for our system. Instead, we consider a binary logistic regression model, which is a special case of the general linear model, yet it does not impose strict assumptions on the distributions of the independent variables. The random component for the (success, failure) outcomes has a binomial distribution [24]. The logit of the probability of success in outcome (i.e., being a flasher in our study) is expressed by a linear function of continuous or categorical predictors. The logit model is $\log \frac{p(x)}{1-p(x)} = \alpha + \beta(x)$.

Our previous work has successfully established the connection between video chat users' skin exposure and being flasher or not. EMeralD defines 3 different variables of Skin Proportion 1, 2, and 3 ($SP_1$, $SP_2$, $SP_3$) to represent users' skin exposure percentages captured by 3 different skin-color spaces. In this paper, we extend the model further by integrating facial features in identifying misbehaving users. We use the 10 discriminative characteristics (facial features) introduced in Section 4.1 as ordinal variables. The value of a variable is dependent upon how the corresponding discriminative characteristic is presented in user's snapshots. For example, the value of variable $NoseMouth[n]$ is 2 if $NoseMouth[2]$ is presented in user's snapshots. To sum up, our theoretical model can be expressed as

$$\log \frac{p(flasher)}{1-p(flasher)} = \alpha + \beta_1 \cdot facialfeature + \beta_2 \cdot skinportion \quad (1)$$

Inter-variable correlations and multicollinearity diagnostic statistics (Variance Inflation Factor - VIF, Tolerance, Condition Index) [25] have been calculated to evaluate the threat of multicollinearity. While there is no formal cutoff value to use with VIF for determining the presence of multicollinearity, values of VIF exceeding 10 and Condition Index exceeding 15 often indicate multicollinearity, but in weaker models, which is often the case in logistic regression, VIF values above 2.5 may be a cause for concern.

Our multicollinearity diagnostic statistics show that multicollinearity threats do exist among several independent variables, namely $FacePos[n]$, $Face[n]$ and three skin portion measures. Common solutions for multicollinearity include dropping one or more correlated variables and combining variables. In this study, we noticed that $FacePos[n]$ and $Face[n]$ have high information overlap – both loaded high on the same dimension, with condition index ex-

[1]The operation of the face classifier has been ignored.

ceeding 15. Since it is neither practical nor meaningful to combine these two ordinal variables, we choose to keep only one of them in the final model to avoid the multicollinearity issue.

When capturing users' exposed skin portions, we apply three aforementioned measures [21]. These three measures provide slightly different information about the degree of skin exposure and are all included in our theoretical model. However, correlation analysis and multicollinearity diagnostics revealed multicollinearities among these measures. To reduce the threats of multicollinearity yet to consider all three measures at the same time, we conduct principal component analysis – a mathematical procedure that transforms multiple correlated continuous variables into a smaller number of uncorrelated variables – on these three predictors before we proceed in the model building process.

We used the principal component analysis procedure in IBM SPSS 19.0 [26] to transform three measures of skin exposure. Kaiser Criterion (Eigen value $> 1$) was followed when selecting components and Scree Plot was used to confirm the dimensions identified. We extracted one component (skinexpcomp) to represent the 3 aforementioned measures of skin exposure according to Eigen values and the elbow point identified in scree plot. When subsequently building our binary logistic regression model, we include only the skin exposure composite (skinexpcomp) which is a linear function of normalized measure scores:

$$skinexpcomp = 0.349 \cdot SP_1 + 0.366 \cdot SP_2 + 0.346 \cdot SP_3 \quad (2)$$

A training sample data was analyzed with the stepwise binary logistic regression procedure in statistical package IBM SPSS 19.0. Maximum Likelihood Estimation with EM algorithm was utilized to estimate the model coefficients. Stepwise logistic regression procedure in SPSS was able to provide multiple models with different combination of independent variables. We identified the optimal model by comparing their goodness of fit indices such as deviance score, AIC (Akaike Information Criterion) and BIC (Bayesian Information Criterion).

# 6. EVALUATION

In this section, we conduct detailed experiments to answer the question of whether our EMeralD system achieves our design goals – both *high accuracy* and *high efficiency* when detecting and filtering misbehaving users in online video chat services. We first compare the accuracy (in terms of precision and recall) of our EMeralD system with state-of-the-art techniques. We then focus on the overall run-time efficiency of EMeralD and its execution cost.

## 6.1 Classifier Performance

In our evaluations, we use a real-world dataset containing 20,000 Chatroulette users' snapshots and randomly split the 20,000 samples into two groups: a 10,000-user training set used to train our pre-classifier and post-classifier; and a 10,000-user testing set to evaluate these classifiers. The 20,000-Chatroulette-user dataset was obtained from Chatroulette system in September 2010 when there were approximately 35% misbehaving users.

**Table 1: Ordering of Association Rules Used in EMeralD**

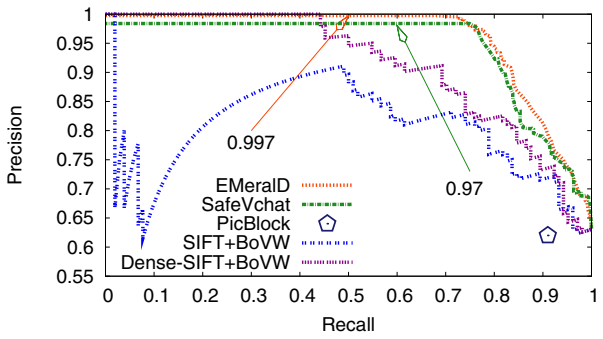
| Order | Association Rule | Confidence |
|---|---|---|
| 1 | $MultiFace[Yes] \Longrightarrow User[normal]$ | 1.00 |
| 2 | $Face[3]\&FacePos[B_2] \Longrightarrow User[normal]$ | 1.00 |
| 3 | $Face[3]\&FacePos[B_3] \Longrightarrow User[normal]$ | 1.00 |
| 4 | $Face[3]\&FacePos[B_4] \Longrightarrow User[normal]$ | 1.00 |
| 5 | $DoubleEye[3] \Longrightarrow User[normal]$ | 1.00 |
| 6 | $DoubleEye[2] \Longrightarrow User[normal]$ | 1.00 |
| 7 | $UpperBody[B_4] \Longrightarrow User[normal]$ | 0.99 |
| 8 | $FaceMouth[3] \Longrightarrow User[normal]$ | 1.00 |
| 9 | $FaceMouth[2] \Longrightarrow User[normal]$ | 0.99 |

We compare the precision and recall of EMeralD with that of the state-of-the-art skin color based detection technique (PicBlock [27]) as well as the integration of SIFT and a Bag-of-Visual-words framework (SIFT+BoVW, Dense-SIFT+BoVW), and our earlier system SafeVchat, which is currently deployed on Chatroulette. As shown in Figure 4, EMeralD significantly outperforms PicBlock [27], because the skin colors in snapshot images captured from online video chat systems are very diverse and the statistical skin-color model used in PicBlock cannot provide effective discriminative characteristics for misbehaving user classification.

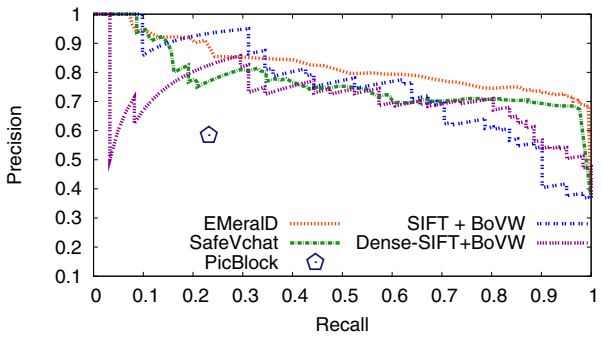
While SIFT+BoVW performs better than PicBlock, it still cannot satisfy the requirement of online video chat systems. Figure 4(a) shows that the recall for classifying normal users is fairly low when the classification precision remains at a high level (e.g., $> 0.95$). As a result, a large number of normal users have to be manually reviewed by human moderators. The main reason behind the poor classification results of SIFT+BoVW is that the SIFT descriptor is a sparse feature representation, which may cause a loss of some discriminative characteristics. To address this issue, we replace the SIFT descriptor with the Dense SIFT descriptor [28] and repeat the experiment of SIFT+BoVW. As shown in Figure 4(a), the recall of Dense-SIFT+BoVW for classifying normal users has increased to 0.50 when precision is high. However, compared with SafeVchat and EMeralD, Dense-SIFT+BoVW still has lower precision and recall, since snapshot images have smaller inter-class distance between normal and misbehaving users.

Finally, we compare the performance between SafeVchat and EMeralD. As shown in Figure 4, the classification performance of EMeralD is slightly higher than that of SafeVchat in terms of precision and recall. Still, this performance improvement is quite beneficial. SafeVchat uses a threshold for the likelihood of being a normal user. This threshold is used to automatically filter out normal users, thus saving human review cost. As shown in Figure 4(a), SafeVchat provides 0.97 precision for classifying normal users with 0.70 recall, while EMeralD reaches 0.997 precision. This indicates that EMeralD only misses (misclassifies) 0.3% of misbehaving users, while automatically and correctly filtering out 70% of normal users – one order of magnitude lower than that of SafeVchat (3% versus 0.3%). To illustrate this improvement, we again take the Chatroulette system as an example. Chatroulette reports that there are approximately 20,000 $\sim$ 40,000 online users at any given time. By using both EMeralD and SafeVchat[2], Chatroulette can automatically filter out about 70% of normal users without involving human reviewers. EMeralD mistakenly leaves only 40 $\sim$ 80 misbehaving users while SafeVchat leaves 400 $\sim$ 800 misbehaving users. Furthermore, we observe that the classification performance of EMeralD for misbehaving users has 5% $\sim$ 8% of improvement on average (see Figure 4(b)). The reasons for these improvements are summarized as follows. (1) The discriminative characteristics that we use in EMeralD is more powerful than the features we used in SafeVchat. One example is that we consider face position as a discriminative characteristic while SafeVchat ignores spatial information. (2) The rule-based pre-classifier provides higher classification capacity for normal users, which dominates the improvement of classification performance. Figure 5 shows that rule-based pre-classifier has high classification precision. The more the rules are harnessed, the higher the classification recall is. (3) Though the classification precision drops when the probabilistic post-classifier is executed, we still observe that our probabilistic post-classifier contributes 6% of recall improvement while maintaining 0.997 precision.

[2]Chatroulette ignores their human review process for the users' snapshots whose likelihoods of being normal users are above 0.97.




(a) Classification results for normal users.




(b) Classification results for misbehaving users.

**Figure 4: Classification performance in terms of precision and recall.**

## 6.2 Run Time Performance

Since PicBlock, SIFT+BoVW and Dense-SIFT+BoVW perform extremely poorly (low precision and recall) in detecting misbehaving users, we have only considered SafeVchat for comparison to evaluate EMeralD's runtime performance. We used EMeralD for two Chatroulette datasets: one is the 20,000-Chatroulette-user dataset that we used for precision and recall evaluation, and the other is a recently collected 30,000-Chatroulette-user dataset that contains 2% misbehaving users. As shown in Table 2, compared to the computation latency of SafeVchat (1276 milliseconds per user), EMeralD system can reduce computation latency by 31.15% on the 20,000-Chatroulette-user dataset. Figure 5 explains the reason behind this. Rule-based pre-classifier is first used to examine users' snapshots. As shown in this figure, the first four association rules successfully filter out 53% of normal users by running only the face classifier of OpenCV. This filtering of 53% of normal users saves the subsequent computation overhead of other OpenCV classifiers. Similar saving is also applied to the other rules except for rule 8 and 9. Note that rule 8 and 9, though correctly filter out 68% of the normal users, do not save computation overhead because the dataset that rule 8 and 9 process has already been examined by all the OpenCV classifiers except the mouth classifier of OpenCV. We can also observe from Table 2 that the latency reduction of EMeralD is affected by the fraction of misbehaving users (i.e., 30,000-Chatroulette-user dataset achieves 48.86% of latency reduction while the other dataset has 31.15% of latency reduction). To further investigate the relationship between the fraction of misbehaving users and the average computation latency, we randomly select 1,000 users from the 20,000-user dataset and manually tune the fraction of misbehaving users. We observe that the average computation latency per user has a linear relationship with the fraction of misbehaving users (Figure 6). The higher the fraction of misbehaving users, the more is the computation latency. The reason behind this is quite straightforward – misbehaving users usually have to be examined by the probabilistic model which uses the outputs of all OpenCV facial feature classifiers which are computation intensive. Another surprising observation from Table 2 is that the average computation latency reduction per user can reach 79.45% when the confidence threshold (minimum confidence) that is used for association rule selection in the pre-classifier is decreased from 99% to 97%. Figure 6 indicates that lower confidence threshold typically results in lower computation latency, because more association rules will be selected to use in the pre-classifier of EMeralD, which makes more users filtered out and labeled as normal users. Note however the decrease in the confidence threshold also gives rise to the decrease in classification precision of the pre-classifier.

**Table 2: Comparison of Computation Latency**

| Confidence | Chatroulette dataset | 30,000 users | 20,000 users |
|---|---|---|---|
| | % of Misbehaving users | 2% | 35% |
| 0.99 | Avg. latency per user | 653 ms | 878 ms |
| | Latency reduction | 48.86% | 31.15% |
| 0.97 | Avg. latency per user | 262 ms | 782 ms |
| | Latency reduction | 79.45% | 38.67% |

**Table 3: Performance Comparison of EMeralD and SafeVchat**

| | # of instances | Throughput per instance | Total cost per month |
|---|---|---|---|
| EMeralD | 57 | 190,350 snapshots per hour | $5,089 |
| SafeVchat | 182 | 59,424 snapshots per hour | $16,249 |

To test how EMerald performs on a large scale, we deployed EMeralD on 57 large instances on Amazon's EC2 infrastructure. These server instances were driven by a large collection of PlanetLab clients generating image snapshots to emulate the Chatroulette workload of about 11 million snapshots per hour. Each instance that the EMeralD code executed on had 7.5 GB of memory and 4 EC2 Compute Units, running a 64-bit version of SUSE Linux Enterprise 11. Each instance processes the snapshots of $8 \sim 10$ Chatroulette users in parallel with $85.32\% \sim 93.01\%$ of CPU utilization. Using this deployment, we on average process 190,350 Chatroulette snapshots per hour for every instance. As shown in Table 3, this throughput is three times that of SafeVchat, which has 59,424 snapshots per hour for every instance, and used 182 EC2 instances as of December 2010. This three-fold improvement in throughput at a large scale mainly results from the reduction of computational complexity.

We also estimate the cost savings of using EMeralD in Table 3. Under current cost assumptions at Amazon for large VM instances, we estimate that EMeralD would only cost Chatroulette $5,089 per month in computation, whereas we estimate SafeVchat currently costs Chatroulette about $16,249 per month for the current user workload, generating 11 million snapshots per month. Note that though tens of millions of snapshots stream into Chatroulette, extra storage instances are not necessary because the size of a user's snapshot is relatively small. Every 15,000 snapshots need approximately 100 MB of storage space and a large instance has 850 GB of local storage. 57 large instances can store users' snapshots for a month without being overwritten. In practice, Chatroulette only keeps users' snapshots for several hours. Therefore, this cost analysis only involves the cost of computation resources for Chatroulette.

# 7. CONCLUSION

We have presented EMeralD, a novel system for misbehavior detection in online video chat systems that substantially improves scalability while improving accuracy. In EMeralD's two-stage approach, first a rule-based pre-classifier identifies and filters out nor-

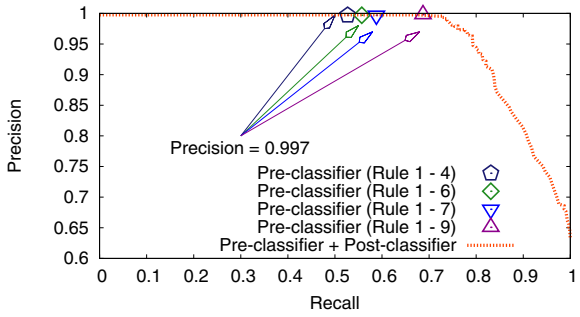



**Figure 5: Classification performance for individual classification components.**

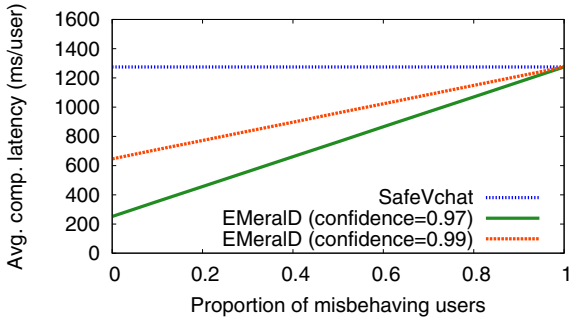



**Figure 6: Proportion variance of misbehaving users versus computation latency of EMeralD.**

mal users by executing only as many classifiers as are needed to reach a probability threshold for identification, thus saving on needless computation. Moreover, the rules are reordered using an A* search algorithm to minimize latency. In stage two, remaining users are processed by a post-classifier using binary logistic regression to accurately identify misbehaving users. Using the real-world image datasets obtained from Chatroulette, we demonstrate compared to prior work that (1) EMeralD achieves improved precision and recall of identifying misbehaving users, (2) EMeralD lowers the computational latency by at least 31-49%, and (3) EMeralD scalably achieves 3 times higher throughput on a large scale Amazon EC2 server deployment.

## Acknowledgment

We thank Andrey Ternovskiy, the founder of Chatroulette.com, for providing us with these otherwise unobtainable internal data traces. We also thank Wenke Lee for providing computation resources in Georgia Institute of Technology. This work is supported in part by the unrestricted gift funds from Chatroulette and the US National Science Foundation (NSF) through grant number CiC 1048298.

## 8. REFERENCES

[1] "Chatroulette web site," http://www.chatroulette.com/.
[2] "myYearbook live web site," http://live.myyearbook.com/.
[3] "Omegle web site," http://www.omegle.com/.
[4] "Tinychat web site," http://tinychat.com/.
[5] "Chatroulette parts with private parts, looking for a new look," http://techcrunch.com/2011/05/04/chatroulette-parts-with-private-parts-looking-for-a-new-look/.
[6] "Can Chatroulette get it up again?" http://techcrunch.com/2010/08/23/chatroulette-up-again/.
[7] "Flasher detection algorithm aims to clean up video chat," http://www.technologyreview.com/blog/arxiv/26281/.
[8] "YouTube web site," http://www.youtube.com/.
[9] "Flag violations and manual inspection on YouTube," http://www.google.com/support/youtube/bin/answer.py?answer=118747.
[10] M. J. Jones and J. M. Rehg, "Statistical color models with application to skin detection," in *International Journal of Computer Vision*, 1999, pp. 274–280.
[11] M. Hammami, Y. Chahir, and L. Chen, "Webguard: A web filtering engine combining textual, structural, and visual content-based analysis," *IEEE Trans. on Knowl. and Data Eng.*, vol. 18, pp. 272–284, 2006.
[12] W. Hu, O. Wu, Z. Chen, Z. Fu, and S. Maybank, "Recognition of pornographic web pages by classifying texts and images," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1019–1034, 2007.
[13] H. Zheng, M. Daoudi, and B. Jedynak, "Blocking adult images based on statistical skin detection," in *Electronic Letters on Comp. Vision & Image Analysis*, 2004, pp. 1–14.
[14] J.-S. Lee, Y.-M. Kuo, P.-C. Chung, and E.-L. Chen, "Naked image detection based on adaptive and extensible skin color model," *Pattern Recog.*, vol. 40, pp. 2261–2270, 2007.
[15] P. Kakumanu, S. Makrogiannis, and N. Bourbakis, "A survey of skin-color modeling and detection methods," *Pattern Recog.*, 2007.
[16] C. Jansohn, A. Ulges, and T. M. Breuel, "Detecting pornographic video content by combining image features with motion information," in *Multimedia*, 2009.
[17] T. Deselaers, L. Pimenidis, and H. Ney, "Bag of visual words models for adult image classification and filtering," in *ICPR*, 2008, pp. 1–4.
[18] A. Lopes, S. Avila, and A. Peixoto, "A bag-of-features approach based on hue-sift descriptor for nude detection," in *Proc. of the 17th European Signal Processing Conf.*, 2009, pp. 1552–1556.
[19] D. Lowe, "Object recognition from local scale-invariant features," in *Proc. of Int'l Conf. on Computer Vision*, vol. 2, 1999, pp. 1150–1157.
[20] K. Van De Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, pp. 1582–1596, 2009.
[21] X. Xing, Y. Liang, H. Cheng, J. Dang, S. Huang, R. Han, X. Liu, Q. Lv, and S. Mishra, "SafeVchat: Detecting obscene content and misbehaving users in online video chat services," in *WWW*, 2011.
[22] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *VLDB*, 1994.
[23] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. on Systems, Science, and Cybernetics*, 1968.
[24] A. Agresti, *Analysis of Ordinal Categorical Data*. Wiley, 1984.
[25] "Detection of multicollinearity," http://en.wikipedia.org/wiki/Multicollinearity.
[26] "IBM SPSS," http://www-01.ibm.com/software/analytics/spss/.
[27] "Picblock," http://www.cinchworks.com/.
[28] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int'l J. of Computer Vision*, pp. 91–110, 2004.