Improved Open-World Fingerprinting Increases Threat to Streaming Video Privacy but Realistic Scenarios Remain Difficult

Timothy Walsh
Naval Postgraduate School
Monterey, California, USA
timothy.walsh@nps.edu

Armon Barton
Naval Postgraduate School
Monterey, California, USA
armon.barton@nps.edu

Mathias Kölsch
Naval Postgraduate School
Monterey, California, USA
kolsch@nps.edu

Abstract

Recent work on video stream fingerprinting has begun to explore its effectiveness in large open-world scenarios, in which the vast majority of test samples are from unmonitored videos that are unknown to the model at training time, showing that it is more difficult than earlier small open-world results have suggested. However, the evaluated approach employed deep learning techniques with potential shortcomings for the open-world task. We build on that work to evaluate more advanced techniques drawn from the literature on open set recognition, out-of-distribution detection, and robustness to adversarial examples, hypothesizing that they can improve effectiveness. We find that combinations of techniques can improve effectiveness, cutting the open-world false positive rate by up to 92% at a recall of 0.5. However, precision would likely still be problematic at the full-scale of the largest platforms hosting hundreds of millions or more videos. Additionally, we find that introducing two other dimensions of realism – when training and test sets are streamed from different vantage points, and when monitoring shorter videos or traffic flows – can greatly increase open-world false positives, making the full-scale open-world task even more difficult. Accordingly, we call for more work to focus on larger and more realistic open-world scenarios to continue to gain a better understanding of the effective envelope for fingerprinting.

Keywords

traffic analysis, streaming video, Tor, neural networks, open set recognition, out-of-distribution detection, adversarial robustness

1 Introduction

End-to-end encryption hides the payloads in network packets from passive adversaries, but it does not hide the patterns in sequences of packet sizes, directions, and timing. Network traffic fingerprinting exploits that information leakage and has called into question the security of encrypted web browsing [5, 23, 44], Domain Name System (DNS) queries [55], web search queries [47], voice communications [65], and streaming video [11, 24] even with the added protection of Tor [62].

Video stream fingerprinting intends to recognize what video a user has watched or is watching by observing their network traffic and comparing it to examples of traffic collected when streaming known videos. Whereas most of the literature has been limited to closed-world or small open-world scenarios, which can be insightful and relevant for some purposes, we take the position that full-scale open-world scenarios are deserving of more consideration and analysis. Some recent work has begun to consider full-scale open-world scenarios [62], showing the difficulty of the task compared to smaller scenarios, but those evaluations have been limited to testing deep learning models that employed only the simplest techniques for the task of classifying samples as being from monitored (known) vs. unmonitored (unknown) videos. We continue this line of work by asking, can open-world video stream fingerprinting become more effective by leveraging more advanced deep learning techniques drawn from the literature on open set recognition (OSR), out-of-distribution detection, and robustness to adversarial examples?

Unanswered questions about the effectiveness of video stream fingerprinting also go beyond whether more advanced techniques can improve precision in open-world scenarios of a larger size. Previous experiments often contain the implicit and unrealistic assumption that training and test sets can be streamed and collected from the same host on the same network. Those that include a more realistic evaluation of models trained and tested under different network conditions still do so from generally the same geographic vantage point, or by artificially altering either the network conditions or traffic traces. Additionally, the results have been mixed and none have included a large open-world evaluation. Similarly, there have been many previous evaluations of effectiveness when an adversary is constrained to fingerprinting shorter video traffic flows, but only in closed-world scenarios. These gaps motivate the need for more rigorous open-world evaluations of these potentially realistic scenarios too as we aim to better understand the threat.

1.1 Contributions

(1) We adapt a variety of deep learning techniques – drawn from the literature on OSR, out-of-distribution detection, and robustness to adversarial examples – to open-world network traffic fingerprinting for the first time and compare their effectiveness with the baseline approach.
(2) We find that novel combinations of Bayesian methods, data augmentation with *mixup*, and none-of-the-above (NOTA) defensive padding yielded our greatest improvements over the baseline approach.
(3) We reason about the significance of the improvements, arguing that they may still fall short of threatening the largest platforms at full-scale, and that Tor still provides a degree of protection compared to streaming video without Tor.

Proceedings on Privacy Enhancing Technologies 2025(4), 130–145

https://doi.org/10.56553/popets-2025-0123

(4) We show the effects of training a model and employing it to recognize the same videos streamed from different geographic vantage points. We find that across some pairs of vantage points, the open-world false positive rate (FPR) at 0.5 recall increased by up to 50x even when closed-world accuracy remained above 0.99.
(5) We show the effects of different video traffic flow lengths on open-world FPR, and we describe how the difficulty of open-world fingerprinting is compounded when monitoring shorter traffic flows due to the real-world distribution of video lengths on the web.

1.2 Ethical Principles

This research did not involve any human subjects or real user data. As such, we did not need to obtain informed consent, apply any methods to preserve privacy, or submit to a review by an ethics panel. This research also does not disclose nor exploit any previously unknown system vulnerabilities. By seeking to better measure the extent of a known vulnerability, we believe that we can advance the science of security and privacy for the benefit of society.

2 Related Work

Numerous works have demonstrated the ability to fingerprint streaming video traffic with various methods and under various conditions, but most have been limited to closed-world scenarios with small datasets of between ten and 100 different videos [1–3, 10, 11, 14, 21, 24, 25, 30, 32, 38, 50, 66]. Reed et al. [51] and Björklund et al. [9] demonstrated high accuracy in larger but still closed-world scenarios. Fewer works have shown open-world results and most test sets have contained fewer than 2,000 different unmonitored videos [4, 13, 15, 54, 69].

Walsh et al. [62] explored video stream fingerprinting in the largest open-world scenarios to date, seeking to better understand effectiveness at the full-scale of platforms that host hundreds of millions or more videos. The authors collected and made available a new dataset including test sets with up to 64,000 different unmonitored videos, streamed over both Tor and Hypertext Transport Protocol Secure (HTTPS)-only connections. They found that it was possible to reproduce the high recall and precision of earlier works when the world size was small but that, even at low rates of recall for the monitored videos, FPRs converged to non-zero values as the world size (i.e. number of different unmonitored videos) grew. They offered a preliminary conclusion that extrapolating such FPRs would make the approach ineffective at the full-scale of the largest video hosting platforms due to the base rate problem, which Juarez et al. [29] similarly argued in the context of open-world Tor website fingerprinting. However, Walsh et al. also suggested that more recent advancements in deep learning techniques specifically for the open-world task could improve an adversary's results, and we explore that possibility using their same dataset.

Dahanayaka et al. [13] were the first to relate open-world network traffic fingerprinting to the general task of OSR. They compared simple baseline approaches with two more advanced OSR techniques, OpenMax [8] and a variation of OpenMax called k-Logit Neighbor Distance (k-LND), for open-world video stream fingerprinting. However, their test sets were very small, containing only six to twelve different unmonitored videos. Additionally, their results were mixed and more recent techniques in the general OSR literature have significantly outperformed OpenMax. We select several of those more recent techniques and compare their performance to the baseline on the largest open-world test sets.

To our knowledge, there is no existing literature that frames the general OSR task as being related to achieving robustness against adversarial examples [20, 59]. We hypothesize that there is a relationship and experiment with two simple but effective techniques drawn from the literature on adversarial robustness: *mixup* by Zhang et al. [67] and NOTA defensive padding by Barton [6] and Jatho [28].

In the area of open-world Tor website and same-domain subpage fingerprinting, Wang [63] argued that what is needed to properly assess open-world fingerprinting performance is a version of precision, π_r, that incorporates the base rate of monitored samples in the wild, which we use to further assess the significance of our empirical results. Wang also tested several "precision optimizers" to improve the open-world effectiveness of Tor website fingerprinting systems using traditional machine learning methods, which Mathews et al. [41] later adapted to deep learning methods. We mirror their efforts by testing a different set of techniques to improve the existing deep learning approach to open-world video stream fingerprinting.

In the context of Tor website fingerprinting, Juarez et al. [29] and Oh et al. [48] considered the question of different geographic vantage points and network conditions. Juarez et al. collected data on machines in three different countries, and then trained different models on the data from each country while testing them on the data from the other countries. The results were mixed, so the authors identified this as a possible limitation of fingerprinting attacks. Oh et al. considered the use of different Tor circuits that they characterized as being either fast or slow and found that this made little or no difference to model performance. However, streaming video is likely more sensitive to network conditions along the path between the client and server. In addition to the first-order effect of requests and responses exhibiting different timing patterns, the use of Dynamic Adaptive Streaming over HTTP (DASH) causes clients to request entirely different responses (i.e. video segments at higher or lower quality levels) at the application layer as a second-order effect in response to different network conditions. Dubin et al. [15] experimented with adding artificial delays and packet loss to the test set to see how the performance of their models degraded, simulating different network conditions for training and testing. Carlson et al. [11] similarly explored performance in closed-world scenarios while scaling and varying client bandwidth. Instead of introducing artificial network condition variability, Schuster et al. [54] and Zhang et al. [69] experimented on real traffic from a wired university campus network vs. real traffic from a wireless residential network. Again, results have been mixed, and none have been shown yet for large open-world scenarios. While Schuster et al. and Zhang et al. collected data from different networks, rather than collecting data from the same network and artificially adding variability, their clients were also located in the same city that likely

shared much of the same path to the same servers. In contrast, we consider vantage points on different continents as an extreme test of a model's ability to generalize.

Among the many previous works that have analyzed shorter vs. longer video traffic flows [4, 9, 11, 14, 21, 24, 30, 50, 51, 69], there is agreement that longer traffic flows can be classified with greater accuracy, presumably due to containing more fingerprintable features. With that said, all previous evaluations have been in closed-world scenarios so it is difficult to reason from those results about the magnitude of the effect in a large open-world scenario. Furthermore, we hypothesize that the real distribution of video lengths on the web compounds the difficulty of recognizing shorter lengths of video in an open-world scenario, and we show that for the first time.

3 Threat Model

Here we define the threat model of interest to us and justify why it deserves further exploration, reasoning about a spectrum of world sizes based partly on an adversary's prior knowledge.

We consider a passive, local, network-level eavesdropper – such as an Internet service provider (ISP) that could be malicious itself or compromised by another malicious entity – engaged in dragnet surveillance to identify viewers of monitored content. The eavesdropper is local in the sense that, when fingerprinting Tor traffic, it is located between the client and the entry relay so that it can still see the client's Internet Protocol (IP) address. Also in the case of fingerprinting Tor traffic, we assume that the adversary can first identify visits to video hosting platforms (e.g. YouTube, Vimeo) through website fingerprinting. We acknowledge that Tor website fingerprinting remains an active area of research, but some of the most recent literature has argued that it is realistic to infer whether a Tor user has visited popular websites under certain conditions with reasonable precision [12, 44]. There are also techniques for separating video traffic in general from a mix of other types of traffic for the purposes of network management [37, 39, 40, 68], and these can also be used to first filter out non-video traffic.

The *world size* is the number of different videos that a user could view from a given hosting platform. It is a function of the number of videos hosted in the catalog, and any prior knowledge that the adversary may have to reduce that number. A platform may host billions of videos, but an adversary could know that surveilled users will only choose from a small selection of those videos. If that smaller number is small enough, the adversary's task can be a closed-world problem. At the extreme, there is even a trivial case in which the adversary knows exactly which video a user will view in advance. Towards the other end of the spectrum, with minimal prior knowledge to reduce the world size, if the number of videos is very large or growing faster than an adversary can obtain known samples and train a model, the task necessarily becomes an open-world problem. The threat model, then, must include an assumption about the adversary's prior knowledge, and we assume none.

3.1 On Assumptions of Prior Knowledge

We assume no prior knowledge to reduce the world size for two reasons. First, while scenarios in which the adversary has a high degree of prior knowledge (e.g. reducing YouTube to as few as ten videos) are relatively well explored in the literature, scenarios on the other end of the spectrum have received much less attention. Second and more importantly, assuming prior knowledge is problematic because it raises unanswered questions of how much is realistic, and how an adversary would realistically obtain it, especially for the task of fingerprinting videos vs. websites, and especially in a dragnet scenario.

One might begin by considering that some content is much more popular than other content. Less than 4% of YouTube videos have accounted for almost 94% of all views [43], so one could focus on fingerprinting only those most popular videos, accepting false positives (FPs) on the rest but potentially still achieving high precision. Indeed, in their study of Tor website fingerprinting, Cherubin et al. [12] similarly found that visits to websites by real users formed a power law distribution, and that they could attain reasonably high recall and precision when monitoring five of the most popular websites. One problem with this is that the sensitive videos and websites that an adversary would care to monitor might not be among the most popular. Another problem for videos is the sheer number of them, with YouTube hosting ten billion as of 2022 [43] so that even 4% of its catalog would still number in the hundreds of millions.

The growth rates of video hosting platforms and the fleeting nature of video popularity pose more problems for fingerprinting videos. YouTube receives billions of uploads per year [43], and Vimeo has reported 350,000 uploads per day [61]. Whereas the daily popularity of websites appears to be quite stable – e.g. google.com, youtube.com, and facebook.com have been the three most visited website domains continuously since 2012 [58] – the most popular videos change daily if not hourly. The most popular videos over the next day are likely to be ones with very few views so far, because they have just been uploaded, or ones that have not even been uploaded yet and thus cannot be known at the time of training a model for future deployment.

If surveilling users in a specific region of the world, for example, one might also consider ruling out all uncommon foreign language content, reasoning that such content would be on the long tail of the distribution and safely ignored. While this might be true in the aggregate, it is not necessarily true for the viewing habits of any individual user that might be caught up in the dragnet. If an adversary targets a specific individual, it could be reasonable to rule out vast swaths of unmonitored content as a possible match for observed traffic, but for dragnet surveillance we believe that the most realistic model is one that assumes an equal prior probability for all videos – i.e. that videos appear in the wild at any given time with a uniform distribution – and therefore does not reduce the world size.

3.2 On the Utility of Closed-World Analyses

Despite the above arguments, we acknowledge that closed-world analyses still have utility. Surveillance targeting an individual user could be informed by significant prior knowledge to greatly shrink the world size. For platforms that serve relatively small (i.e. thousands of videos) and slowly growing catalogs of curated film and television content (e.g. Netflix, SVT Play), as opposed to platforms

that host user-uploaded content, Reed et al. [51] and Bjorklund et al. [9] showed that it was possible to treat them as closed-worlds. When proposing and evaluating defenses against fingerprinting, success in a small closed-world scenario can provide good evidence that a defense is effective, like a cryptographic proof of security showing that two encrypted messages are computationally indistinguishable. Finally, as a proof-of-concept for novel fingerprinting techniques, even very small closed-world analyses can show effectiveness relative to existing techniques.

4 Improving Large-Scale Open-World Fingerprinting

Here we evaluate more advanced deep learning techniques to see if they can improve the effectiveness of large-scale open-world fingerprinting. We begin with some background on the task of open-world fingerprinting, its relationship with the general task of OSR, and the baseline approach. We then explain selected techniques that we hypothesize can improve upon the baseline, and our implementation of them. Finally, we discuss our experimental results. We find that we can improve upon the baseline, but also that the task remains difficult.

4.1 Connection Between Open-World Fingerprinting and OSR

In the open-world fingerprinting scenario, users can view pages or videos that are in the adversary's *monitored* set, of which he has known samples at training time and is trying to recognize, or from perhaps billions of other pages or videos called the *unmonitored* set. In general, the machine learning literature calls this task OSR [19, 52, 53] and it is an active area of research, closely related to anomaly detection and out-of-distribution detection.

Hendrycks et al. [26] proposed a baseline approach to OSR for neural networks using maximum softmax probability (MSP). The intuition is that, given a test sample and a model's predicted probabilities for each known class, "Correctly classified examples tend to have greater maximum softmax probabilities than erroneously classified and out-of-distribution examples, allowing for their detection" [26]. They then showed that this was unreliable across benchmark datasets for computer vision, speech recognition, and text classification. Despite these findings, MSP is consistent with the existing approach to open-world video stream fingerprinting with deep neural networks [4, 13, 54, 62].

4.2 Baseline Approach with MSP and the Standard Model

Successful use of MSP depends on well-regularized decision boundaries between the N known classes so that new samples of known classes at test time will be farther from the boundaries, and new samples of unknown classes will be closer to the boundaries. Figure 1a illustrates the intuition for this technique in a simplified, conceptual scenario with samples in two-dimensional space. Samples of the monitored video classes form clusters. However, complex neural networks can find boundaries that overfit the training data or are overly complex in regions where training data is sparse, such as between the clusters of samples representing the known classes. Figure 1b illustrates a scenario in which the model learns decision boundaries that still minimize the loss during training but might fail to generalize well at test time.

In an attempt to improve model behavior in the regions of hyperspace around and between the monitored video classes, the existing baseline approach to open-world fingerprinting includes randomly drawn unmonitored video samples in the training set with an additional $N + 1$ class label; coined the "Standard Model" [56] in Tor website fingerprinting and "known unknowns" [19] in the general OSR literature. The idea is to explicitly teach the model the distribution of unmonitored videos and draw in tighter decision boundaries around the monitored video classes. However, this still may not optimally seed the open space to produce decision boundaries that generalize well. The random sampling of unmonitored videos cannot fully represent other unmonitored videos ("unknown unknowns" [19]), so a new unmonitored sample may still be well inside the decision boundary for a monitored video class, resulting in a FP, as illustrated in Figure 1c.

Another issue is the tendency of large and complex neural networks to make predictions that are poorly calibrated, where the predicted probability for a given class tends to be much higher than the frequency of that being the true class, making it hard to separate in-distribution from out-of-distribution predictions [22, 26]. Finally, in addition to decision boundaries being irregular, the predicted probabilities can also change sharply around these boundaries, increasing a model's sensitivity to small input changes and susceptibility to adversarial examples [20, 59].

The potential improvements that we explore aim to address the aforementioned theoretical drawbacks in distinct ways.

4.3 More Advanced Techniques

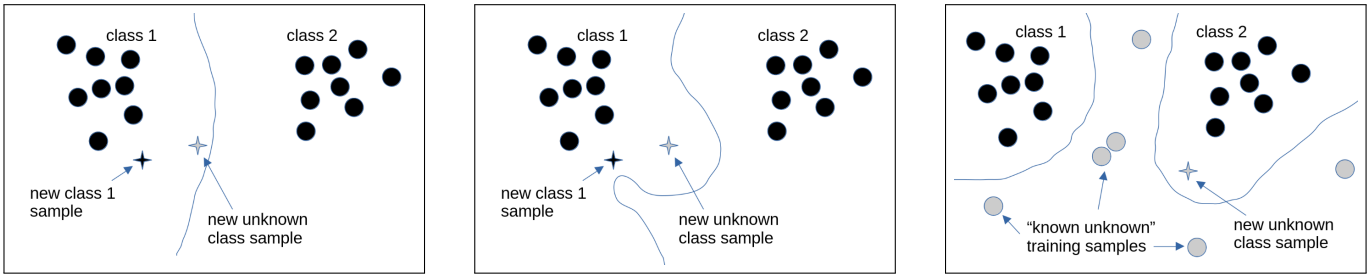
4.3.1 Bayesian Methods. The underlying concept of Bayesian machine learning methods is that there is a true posterior probability distribution of model parameters given data, $P(\theta|D)$, and we can express it using Bayes' Rule. This is a function of the likelihood of the data given the model parameters, $P(D|\theta)$, an assumed prior probability distribution for the model parameters, $P(\theta)$, and the probability of the data, $P(D)$:

$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Bayesian methods involve finding a distribution to approximate the true $P(\theta|D)$ instead of finding just a single set of model parameters that are a point estimate. A good approximate distribution enables us to draw a number of models with distinct parameters that are good for both the data and an assumed prior. As with any ensemble of models, we can then obtain a number of distinct predictions. This can be advantageous in several ways.

First, taking the average prediction of an ensemble can improve accuracy [36]. Drawing m Monte Carlo samples from a distribution that approximates $P(\theta|D)$, followed by Bayesian model averaging of the predictions, is described mathematically as

$$p(y|x, D) \approx \frac{1}{m} \sum_{i=1}^{m} p(y|x, \theta^i) \text{ where } \theta^i \sim p(\theta|D)$$



(a) MSP intuition **(b) MSP failure case** **(c) Standard model**

Figure 1: Intuition for MSP (a), a case in which MSP fails (b), and the Standard Model (c). Samples from known or monitored classes are black. Samples from unknown or unmonitored classes are gray.

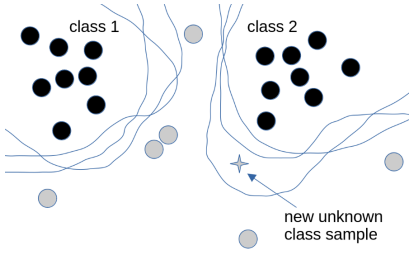


Figure 2: Intuition for using a Bayesian model average. Between known classes where training data is sparse, the Bayesian model average expresses more uncertainty.

Second, Bayesian model averaging can yield predictions that are not just more accurate but also better calibrated [36]. Even if all of the models agree on the predicted class, they will do so with varying probability. The average probabilities can better align with the frequencies of being correct.

Third, different predictions allow us to capture more uncertainty information. We can measure total uncertainty as the entropy of the predicted categorical distribution, $H(y|x, D)$, from a single model or a Bayesian model average. Total uncertainty is the sum of aleatoric and epistemic uncertainty. Aleatoric uncertainty is due to the inherent noisiness of training data. Epistemic uncertainty is due to different models disagreeing in their predictions for a given sample. For a single model there is no disagreement and therefore no epistemic uncertainty component, so a Bayesian model average provides additional uncertainty information. Epistemic uncertainty for a given sample can arise from models not having seen enough representative training data on which to base their predictions. This might be indicative of a sample belonging to an unmonitored class in our open-world fingerprinting scenario that was unknown at training time. Figure 2 illustrates the intuition for a Bayesian model average being more reliable than any one deterministic model.

A distribution to approximate the true $P(\theta|D)$ is typically called a variational distribution, $Q_\phi(\theta)$, and it can be found through variational inference. The objective in variational inference is to find parameters ϕ to minimize the Kullback-Leibler divergence (KLD) between $Q_\phi(\theta)$ and $P(\theta|D)$. Gal and Ghahramani [16] showed that training with dropout can approximate variational inference, and proposed Monte Carlo dropout (MCD) as a method to use an already trained model in a Bayesian manner. MCD has the advantage of being easy to implement, but the quality of the uncertainty information is limited due to sampling only from a Bernoulli distribution for each neuron, and due to the dropout rate being a fixed hyperparameter.

A more theoretically sound but computationally expensive Bayesian method called Spike-and-Slab Dropout [42] combines dropout and Gaussian variational inference. This assumes a variational distribution that is a product of Bernoulli distributions for the neurons and Gaussian distributions for the parameter weights. To improve upon the dropout rate being a fixed hyperparameter for each layer of neurons, another technique called Concrete Dropout [17, 31] makes the rate learnable through backpropagation.

To efficiently draw different weights for each sample during training, we used convolutional and linear flipout [35, 64] layers. We set the prior for the dropout rate to 0.5 to reflect maximum uncertainty about the optimal value while preventing it from trivially collapsing to 0.0 to optimally fit the data. At test time, for each sample, we drew from our variational distribution ten times before computing the Bayesian model average and using the resulting MSP or total uncertainty to rank predictions. Using the MSP is straightforward. Ranking predictions by uncertainty was less straightforward, and we describe the problem along with our solution in Appendix A.

4.3.2 Data Augmentation with mixup. As we already described, deep neural networks can learn decision boundaries that are irregular in some regions of input or feature space where training data is sparse. The predicted class probabilities can also change sharply around these decision boundaries. This can result in errors due to a model's output being sensitive to small changes to inputs. An adversarial example is a sample of one class that is intentionally perturbed in ways that are nearly imperceptible (to the human eye in the traditional setting of computer vision) to take advantage of such sensitivity and induce the model to predict a different, incorrect class [20, 59].

To better regularize neural networks and gain robustness against adversarial examples, Zhang et al. [67] proposed a technique called *mixup*. The technique aims to make the model's behavior in the space between training data more linear. For each batch during training, it pairs each real sample x_i with another real sample x_j

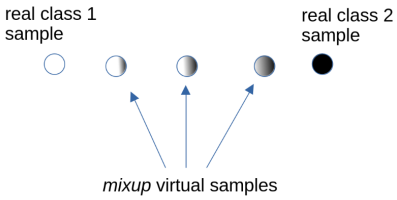


Figure 3: Intuition for using *mixup* to encourage more linear behavior of the model between classes.

from the batch at random. It then computes a linear interpolation between both the input features x_i, x_j and the corresponding labels y_i, y_j. The interpolation is weighted by a value of λ, between 0.0 and 1.0, drawn randomly from a beta distribution for each batch. The beta distribution is defined by a hyperparameter, α. The resulting virtual samples ($\tilde{\mathbf{x}}$,$\tilde{\mathbf{y}}$) are what the model trains on. We illustrate this in Figure 3.

$$\lambda \sim \text{Beta}(\alpha, \alpha)$$
$$\tilde{\mathbf{x}} = \lambda \mathbf{x}_i + (1-\lambda)\mathbf{x}_j$$
$$\tilde{\mathbf{y}} = \lambda \mathbf{y}_i + (1-\lambda)\mathbf{y}_j$$

While there are no adversarial examples in our open-world video stream fingerprinting dataset, we are similarly concerned with the model's sensitivity and behavior in the regions of space near the decision boundaries where we lack training data. There is effectively no difference between an adversarial example and a new sample of an unknown class that happen to have identical and difficult to classify features. In both cases, we want the model to indicate its uncertainty rather than outputting a high probability for a known class. Thulasidasan et al. [60] found that *mixup* improved calibration and out-of-distribution detection even more than MCD. Therefore, we hypothesize that open-world video stream fingerprinting could similarly benefit from the regularizing effect of *mixup*.

We applied *mixup* exactly as shown by Zhang et al., interpolating between random pairs of samples in each batch, from all classes, in input space. We tuned α over the validation sets, grid searching between 0.01 and 0.5 to find what yielded the best cross-entropy loss and area under the curve (AUC) for the binary open-world precision vs. recall.

4.3.3 NOTA Defensive Padding. The desire for robustness to adversarial examples also motivated the development of NOTA defensive padding [6, 28]. As with *mixup*, the intuition is that adversarial examples exist near decision boundaries in regions that are sparse with training data, and therefore where model behavior could be poorly defined. In the case of defensive padding, the idea is to learn better decision boundaries by seeding the open space between classes with virtual training samples. The virtual training samples separate the N known classes (hence the term "padding") and have an $N+1$ class label. This encourages the model to learn that new samples in these regions do not belong to any known class instead of being fooled into predicting the most probable (but incorrect) known class.

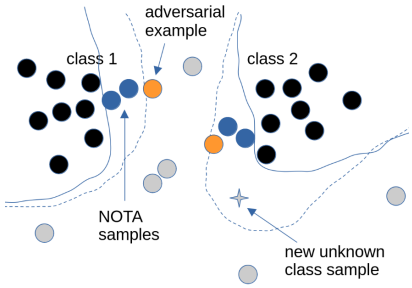


Figure 4: Intuition for augmenting the training set with NOTA defensive padding. Orange points are the adversarial examples created through PGD, and blue points are the mean and uniform padding samples. The dashed line represents the pre-trained baseline model's decision boundaries, and the solid line represents the resulting NOTA-trained model.

Barton [6] first proposed creating virtual training samples by linearly interpolating between random pairs of samples, x_i and x_j, of different classes in each batch. Mean padding samples are the mean of x_i and x_j plus Gaussian noise. Uniform padding weights the interpolation by a value drawn randomly from a uniform distribution. Finally, the virtual training samples for the NOTA class are added back into the batch of original training samples. Jatho [28] later modified the technique by first using Projected Gradient Descent (PGD) to generate an adversarial example x_i' for each x_i, and then creating the mean and uniform NOTA padding between x_i and x_i'. The idea is to place the NOTA samples more closely around each class and force the model to learn even tighter boundaries. Jatho also attained the best results when training a Bayesian model with this technique.

We are similarly interested in forcing the model to learn tight decision boundaries around each monitored video class, not to defend against adversarial examples but against real unmonitored samples that are difficult to distinguish from monitored samples. While real unmonitored training set samples could enable the model to learn the broad distribution of unmonitored videos, we expect that most of those samples would be quite distant and easily distinguishable from the monitored video classes. NOTA samples are instead specifically crafted to be close to the monitored samples and difficult to distinguish, occupying the regions of hyperspace where we believe FPs are most likely to occur. We hypothesize, therefore, that NOTA defensive padding and real unmonitored training samples with the same $N+1$ label can complement each other as illustrated in Figure 4.

We ultimately trained our model on a 1:1:1 ratio of monitored samples, real unmonitored samples, and NOTA samples. To create the NOTA samples, we first generated adversarial examples from monitored samples in each batch, and then linearly interpolated between the monitored samples and the adversarial examples. For the white box model in PGD, we used the baseline model pre-trained to output the softmax probabilities for the $N+1$ classes. We conducted preliminary experiments with both targeted (i.e. driving the white box model's predictions towards the $N+1$ label) and untargeted (i.e. driving them away from the true monitored class label) PGD and

found that our targeted version performed slightly better. Whereas image datasets have well-defined ranges for pixel values and well-understood choices for number of PGD steps, step sizes, and epsilon bounds, our dataset did not and required additional preliminary experimentation. Also, whereas the goal of adversarial examples in computer vision is to fool both the model and the human eye with imperceptible perturbations, our goal is to augment the model's training data with samples that are challenging for the model to classify but do not need to go undetected as adversarial examples. However, it was our intuition that the adversarial examples should still be quite close to the monitored samples and contain only plausible values, otherwise the NOTA samples may not contribute as desired to the model learning tighter decision boundaries. We began by finding the range and standard deviation of the values in each real training sample so that we could define our hyperparameters in those terms. We set our epsilon bound to be a fraction of one standard deviation and set the step size to be the epsilon bound divided by the number of steps, another hyperparameter. Per the usual PGD procedure, we projected the adversarial examples back into the epsilon ball and the range for the real training data after each iteration. For the NOTA mean and uniform padding, we similarly defined the Gaussian noise as a fraction of the standard deviation in the monitored training data. We tuned these hyperparameters with grid searches between 5 and 80 steps, and between 0.00001 and 1.0 standard deviation.

4.3.4 GAN-trained Discriminator. In the context of OSR, noting that randomly drawn training samples of open classes (i.e. the Standard Model) "are unlikely to exhaustively span the open-world," Kong and Ramanan explored the idea to "augment the available set of real open training examples with adversarially synthesized fake data" [33]. The intuition is similar to the intuition for NOTA: seeding the open space close to the known classes with virtual training data could result in tighter decision boundaries. The proposed approach, OpenGAN, improved on earlier approaches to using a generative adversarial network (GAN) for OSR [18, 46] with several new ideas. First, the lower layers of a pre-trained model extract features before the discriminator makes its predictions, and the generator likewise produces fake features instead of raw inputs. Second, the objective function includes a hyperparameter, λ_G, to tune the weight given to generated fakes and real open set samples when updating the discriminator.

$$\max_D \min_G \Big(\mathbb{E}_{x \sim p_{\text{data,mon}}} [\log D(x)] + (1 - \lambda_G) \cdot \mathbb{E}_{x \sim p_{\text{data,unmon}}} [\log(1 - D(x))] + \lambda_G \cdot \mathbb{E}_{z \sim \mathcal{N}(0,1)} [\log(1 - D(G(z)))] \Big)$$

Third, training stops based on the discriminator's performance on an OSR validation set, rather than training until the discriminator and generator converge to an equilibrium. Finally, at test time, the trained discriminator's prediction is used directly as the score for each sample.

In addition to gaining robustness against adversarial examples, Zhang et al. also argued that *mixup* could "stabilize GAN training because it acts as a regularizer on the gradients of the discriminator," and that "the smoothness of the discriminator guarantees a stable source of gradient information to the generator" [67]. Therefore, it is natural to run OpenGAN with *mixup* applied to pairs of fake and real samples during each discriminator update.

We conducted preliminary experiments using the code provided by Kong and Ramanan [34] with only minor adaptations to employ the lower layers of our own pre-trained baseline model as the feature extractor. From this first attempt, we observed that the discriminator underperformed the baseline model on the validation set even when we set λ_G to 0.0, where the approaches are theoretically equivalent. To strengthen the discriminator, we adopted an architecture that was identical to the top layers of our baseline model architecture. With this design, when we set $\lambda_G = 0.0$, the trained discriminator matched the performance of our baseline model. However, its performance on the validation set degraded slightly when we introduced fakes into its training. We also saw that the discriminator was rarely fooled by fakes after each update, and the generator's loss failed to improve after the first few epochs. While Kong and Ramanan described the problem of a discriminator becoming useless when GAN training converges, due to high quality fakes being indistinguishable from real samples and confusing the discriminator, it appeared that we had the opposite problem: poor quality fakes in the training set could reduce the discriminator's ability to generalize to real samples. We further experimented with training in input space, using a generator architecture that was previously successful for producing fake 1D network traffic traces [48, 49], and adding feature matching loss. We tuned hyperparameters for λ_G and the weight of adversarial loss vs. the generator's feature matching loss. We ultimately chose to test the feature space implementation with our improved discriminator and $\lambda_G = 0.5$.

4.4 Experimental Setup

We used the same basic experimental setup described by Walsh et al. [62] for their open-world experiments. This includes the same datasets (one for videos streamed with Tor, and one with HTTPS-only) of four-minute long traffic flows collected when streaming videos from Vimeo, using the same training, validation, and test splits. There are 60 monitored videos with 90 samples of each, and more than 76,000 different unmonitored videos with one sample of each. The test sets contain ten samples of each monitored video and 64,000 unmonitored samples. The remaining samples are for training and validation. We also used the same data representation for the traffic flows (bytes sent and received per $\frac{1}{8}$-second time step for the Tor traffic, and $\frac{1}{16}$-second for HTTPS-only), and the same convolutional neural network (CNN) architecture for the baseline model. We refer the reader to the description by Walsh et al. [62] for more details and access to the datasets and code.

We found it useful to categorize techniques as either types of scores produced by distinct types of models:

(1) Deterministic MSP
(2) Bayesian model average MSP
(3) Bayesian model uncertainty
(4) Binary prediction of a GAN discriminator (i.e. OpenGAN)

or ways to compose or augment the training data:

(A) Samples of the monitored videos

Table 1: Mean performance on the HTTPS-only test set. The first row is the baseline and the best results are in bold.

Score Types				Training Data					Performance Metrics		
1. Deterministic MSP	2. Bayes. Avg. MSP	3. Bayes. Uncertainty	4. GAN Discriminator	A. Mon. Samples	B. Unmon. Samples	C. *mixup*	D. NOTA Def. Pad.	E. GAN Fakes	AUC	FPR @ 0.5 Recall	FPR @ 0.9 Recall
✓				✓	✓				0.988	0.0000102	0.0001563
✓				✓	✓	✓			0.994	0.0000031	0.0000547
✓				✓	✓		✓		0.996	0.0000070	0.0000633
✓				✓	✓	✓	✓		**0.997**	**0.0000008**	**0.0000359**
	✓			✓	✓				0.994	0.0000063	0.0000906
		✓		✓	✓				0.995	0.0000078	0.0000797
	✓			✓	✓	✓			0.996	0.0000055	0.0000617
		✓		✓	✓	✓			0.996	0.0000031	0.0000461
	✓			✓	✓		✓		0.993	0.0000063	0.0001102
		✓		✓	✓		✓		0.994	0.0000070	0.0000844
	✓			✓	✓	✓	✓		0.996	0.0000023	0.0000500
		✓		✓	✓	✓	✓		0.996	0.0000016	0.0000406
			✓	✓	✓			✓	0.994	0.0000039	0.0000469
			✓	✓	✓	✓		✓	0.995	0.0000109	0.0000453

Table 2: Mean performance on the Tor test set. The first row is the baseline and the best results are in bold.

Score Types				Training Data					Performance Metrics		
1. Deterministic MSP	2. Bayes. Avg. MSP	3. Bayes. Uncertainty	4. GAN Discriminator	A. Mon. Samples	B. Unmon. Samples	C. *mixup*	D. NOTA Def. Pad.	E. GAN Fakes	AUC	FPR @ 0.5 Recall	FPR @ 0.9 Recall
✓				✓	✓				0.848	0.0000250	0.0141789
✓				✓	✓	✓			0.871	0.0000117	0.0107617
✓				✓	✓		✓		0.852	0.0000383	0.0191188
✓				✓	✓	✓	✓		0.865	0.0000406	0.0115585
	✓			✓	✓				0.849	0.0000227	0.0140359
		✓		✓	✓				0.857	0.0000125	0.0125617
	✓			✓	✓	✓			0.872	**0.0000063**	0.0109791
		✓		✓	✓	✓			**0.876**	0.0000070	**0.0094039**
	✓			✓	✓		✓		0.807	0.0000828	0.0300266
		✓		✓	✓		✓		0.816	0.0000727	0.0250281
	✓			✓	✓	✓	✓		0.810	0.0000594	0.0352163
		✓		✓	✓	✓	✓		0.821	0.0000516	0.0282852
			✓	✓	✓			✓	0.854	0.0000211	0.0144125
			✓	✓	✓	✓		✓	0.861	0.0000195	0.0125257

(B) Samples of random unmonitored videos (i.e. Standard Model)
(C) Virtual samples produced with *mixup*
(D) NOTA defensive padding samples
(E) Fakes from a GAN (i.e. OpenGAN)

It is possible to pair each of the former techniques with any number of the latter techniques, but we experimented only with selected combinations that were most promising based on our analysis of the literature and understanding of them.

We conducted 20 trials for each approach so that we could better assess expected performance. Each trial involved independently training a fresh model using that approach and testing it. The performance across trials varied due to the stochastic or non-deterministic nature of techniques like random weight initialization, batch gradient descent, dropout, variational inference, and data augmentation.

4.5 Results and Discussion

We show the results in Tables 1 and 2 for the HTTPS-only and Tor test sets, respectively. In each table, the top row is the baseline approach using deterministic MSP and the Standard Model. For comparison, we show the AUC for the precision vs. recall curve. We also show the FPRs at 0.5 and 0.9 recall to give a sense of the range of performance; in Section 4.5.1 we further analyze these and other points of recall within this range. At rates of recall below 0.5, it was common to attain zero FPs even across all 20 trials, but this is not indicative of a true FPR that would scale to any world size; the test set is simply not large enough to measure smaller but non-zero FPRs.

We attained our best results with combinations of Bayesian methods, *mixup*, and (in the HTTPS-only case) NOTA defensive padding. Compared to the baseline on the HTTPS-only test set, these techniques cut the FPR by up to 92% at 0.5 recall, and up to 77% at 0.9 recall. On the Tor test set, the improvements were by 75% and 34%, respectively. The common thread among these methods is that they have all previously been shown to improve adversarial robustness, supporting our hypothesis of a linkage between that task and open-world fingerprinting which could extend to OSR in general.

Our NOTA defensive padding and OpenGAN implementations were less effective on the Tor test set. We find this interesting because they similarly aim to generate new training samples that are very close to the monitored training data but with the unmonitored label. We speculate that generating such samples in a helpful way based on the Tor training data is more difficult due to its inherent noisiness, compared to the HTTPS-only training data, which was evident whenever we visualized the data with t-Distributed Stochastic Neighbor Embedding (t-SNE) plots. Further development of the OpenGAN generator for our task may be necessary to fully realize its potential. There is a large body of work on engineering GANs to produce varied and high quality fake 2D images, but relatively little work has gone into refining GANs for other tasks. GANs are notoriously difficult to train, and the main weakness in our OpenGAN implementation appeared to be getting the generator to produce fakes that were sufficiently similar to the real monitored samples.

4.5.1 *Assessing the Significance of the Results.*

Both precision and FPR can be misleading in their own ways. The base rate fallacy captures the way in which FPR can mislead: in the wild, a *seemingly* low FPR can yield a number of FPs that dominates the number of true positives if the base rate for positives is sufficiently low. Precision avoids the base rate fallacy by directly expressing the probability that a positive prediction is a true positive. Precision can mislead or fail to generalize to real-world scenarios, however, if the base rates or balance of classes in the test set are unrealistic. What we really want to know is the *expected* precision of the model in a real-world scenario.

By manipulating Wang's formula for π_r and applying our empirical FPRs, we can estimate the world size (i.e. number of videos in

a hosting platform catalog) at which an adversary can expect a certain recall and precision in a scenario where users select videos to watch in a uniformly random manner (i.e. all videos appear with an equal base rate). The simplifying assumption of uniformly random video selection and equal base rates gives us a reasonable starting point for the sake of analysis, as we discussed in Section 3, but we discuss this more in Section 6.

We start with the formula for base rate adjusted precision, π_r, proposed by Wang [63]:

$$\pi_r = \frac{R_{TP}}{R_{TP} + R_{WP} + r * R_{FP}}$$

where R_{TP} is the recall (true positive rate), R_{WP} is the "wrong positive" rate, r is the ratio of negative (unmonitored) to positive (monitored) samples in the wild, and R_{FP} is the FPR. Wrong positives are correct positive predictions for the binary classification task of monitored vs. unmonitored, but incorrect for the N-way classification task within the monitored set. We count wrong positives as true positives for simplicity because we are focused on the binary classification task and our N-way classification accuracy is very high, so their contribution to the result is negligible.

Wang further defined $r = \frac{N'_N}{N'_P}$ where the numerator is the number of negative (unmonitored) samples appearing in the wild and the denominator is the number of positive (monitored) samples. Our earlier assumption of uniform random video selection makes the base rate for any monitored video equal to any other video, so we can express the world size as $w \approx N'_N + N'_P$ where N'_N is simply the number of unmonitored videos and N'_P is the number of monitored set videos. Substituting this definition of r into Wang's equation and solving for N'_N, we get:

$$N'_N = \frac{R_{TP} * N'_P}{\pi_r * R_{FP}} - \frac{R_{TP} * N'_P}{R_{FP}}$$

Adding the 60 monitored videos would theoretically give us the world size, w, but that is a rounding error for a back-of-the-envelope estimate. We show an array of these calculations across a range of precision and recall points in Tables 3 and 4.

For example, to compute the estimated world size of 42.9M (shown in the bottom left cell of Table 4), from which we could expect to draw a test set and attain 0.5 recall with 0.1 precision using our best empirical FPR of 0.0000063 (taken from Table 2), we plug in the values as follows:

$$42.9M \approx \frac{0.5 * 60}{0.1 * 0.0000063} - \frac{0.5 * 60}{0.0000063}$$

This assumes a similar set of 60 monitored videos, similar conditions and methods for capturing and parsing traffic samples, and the same training strategy for the same model architecture. It also assumes that the test set is a representative sample of the real distribution of videos hosted by Vimeo.

Considering a range of desired precision and recall points is interesting because, in the same way that Wang [63] argued that multiple observations by an adversary can mitigate the problem of low recall, we argue that multiple observations can also mitigate low precision. With every positive prediction for a given user, the probability that all positives are false decreases, so the adversary can gain some confidence that the user has viewed at least one monitored video. With lower expected recall and precision, the adversary would need to make more observations of a given user over a longer period of time, and vice versa.

Table 3: Estimated maximum world sizes (in millions) in which the adversary could sustain a desired precision (P) and recall (R), derived from our best empirical FPRs on the HTTPS-only test set.

R \ P	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.9	13.5	6.0	3.5	2.3	1.5	1.0	0.6	0.4	0.2
0.8	32.5	14.4	8.4	5.4	3.6	2.4	1.5	0.9	0.4
0.7	54.0	24.0	14.0	9.0	6.0	4.0	2.6	1.5	0.7
0.6	140.9	62.6	36.5	23.5	15.7	10.4	6.7	3.9	1.7
0.5	337.5	150.0	87.5	56.3	37.5	25.0	16.1	9.4	4.2

Table 4: Estimated maximum world sizes (in millions) in which the adversary could sustain a desired precision (P) and recall (R), derived from our best empirical FPRs on the Tor test set.

R \ P	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
0.9	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
0.8	0.3	0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1	<0.1
0.7	1.7	0.8	0.5	0.3	0.2	0.1	<0.1	<0.1	<0.1
0.6	8.0	3.5	2.1	1.3	0.9	0.6	0.4	0.2	0.1
0.5	42.9	19.1	11.1	7.1	4.8	3.2	2.0	1.2	0.5

Even with our substantial improvements over the baseline, we can still see the difficulty of fingerprinting at the full-scale of the largest hosting platforms. The adversary could reach a world size in the hundreds of millions (against HTTPS-only traffic) using our best approach only by accepting a recall of 0.5-0.6 and precision of 0.1-0.2 (cells highlighted in pink) and making many observations over time. Because model predictions on different test samples are only conditionally independent given the model parameters, not truly independent, the adversary would still face the non-trivial task of determining how many positive predictions he would need over time to gain a certain degree of confidence that a given user has viewed monitored content.

We also see that using Tor still provides a substantial degree of protection against video stream fingerprinting (compared to using an HTTPS-only connection) even under the strong assumption that an adversary can first recognize visits to a certain video hosting platform through website fingerprinting.

5 Evaluating Across Other Dimensions of Realism

In the previous section we found that more advanced techniques could improve open-world effectiveness, but that the full-scale scenario remains difficult. Here we evaluate the effects of two other dimensions of realism, beyond size, in a large open-world scenario for the first time and find that they can greatly increase open-world

false positives, potentially making the full-scale open-world task even more difficult and necessitating more targeted attacks.

5.1 Different Geographic Vantage Points

In the previous section and in most of the network traffic fingerprinting literature, experimental setups implicitly assume that adversaries can train a model on traffic captured from the same geographic vantage point, if not the same local area network (LAN), as the users under surveillance. However, this could be infeasible or undesirable for an adversary. An adversary who controls a network can deploy hosts to generate and collect a training set either at the network edge nearer to *some* users, or upstream at the gateway through which *all* user traffic passes. In either case, the training set traffic would originate from a vantage point that is not near most users. Figure 5 provides a conceptual illustration of this motivating scenario.

It would be ideal for an adversary to be able to collect just one training set for a model that could generalize to traffic streamed from anywhere in the network, rather than collecting a different training set per user location. Taken to an extreme, if a trained model effectively generalized even across geographic regions and continents, models could be globally portable. Pre-trained models and their training data could also be shared and leveraged among cooperating adversaries to create more effective models. Conversely, if a trained model *cannot* generalize well to video traffic streamed from elsewhere, this could substantially increase the work factor for an adversary, requiring an attack that is more narrowly targeted. We explore this for the first time with video streamed from ten different vantage points around the world, both with and without Tor, and in the context of a large open-world scenario, as a severe test of the null hypothesis that a model can generalize across vantage points.

5.1.1 Experimental Setup. Our experiments included the Vimeo videos streamed from all ten vantage points in the dataset from Walsh et al.[62] The dataset includes 90 samples of each of the 60 monitored videos streamed from each of the ten vantage points. We further split this into 70 samples of each monitored video for training, ten for validation, and ten for testing. Appendix B provides more details about the vantage points.

We first tested closed-world accuracy across all pairs of vantage points. This means that we separately trained a model from each vantage point and then evaluated it on the test sets from that vantage point and each of the nine other vantage points, for a total of 100 train-test pairings. We repeated this for ten trials using the best model architecture and hyperparameters found by Walsh et al. [62] during their earlier investigation of closed-world video stream fingerprinting.

We then tested models in the large open-world scenario described in the previous section. Because the open-world test set only contains unmonitored video samples collected from the us-west-2 vantage point, we ran the models from the nine other vantage points on the test set and compared their performance to the model trained at us-west-2. Since the training sets for the nine other vantage points do not contain any unmonitored video samples, we could not test the baseline nor best improved approaches using the Standard Model in this scenario. We instead used the simpler MSP from a deterministic model trained without any unmonitored video samples, which allowed us to re-use the same models that we trained for the closed-world experiment. This is sufficient to see the *relative* difference in performance between models from different vantage points, even if it does not allow direct comparisons with the best results from the previous section.

Table 5: Open-world results for models trained at each vantage point and evaluated on the test set streamed from us-west-2. Relative to the baseline (where the model was trained at us-west-2), we show the factor of increase to the FPR at 0.5 recall.

Training Vantage Point	HTTPS traffic	Tor traffic
us-east-1	0.0x	1.2x
sa-east-1	7.4x	1.6x
eu-west-2	2.8x	1.0x
eu-central-1	19.2x	0.7x
af-south-1	50.3x	1.8x
eu-north-1	5.8x	0.8x
me-central-1	19.8x	1.1x
ap-northeast-2	7.4x	17.7x
ap-southeast-1	1.0x	25.2x

5.1.2 Results and Discussion. We show open-world results in Table 5. The HTTPS-only experiments yielded our most interesting results. Even though 60-way closed-world accuracy across all pairs of vantage points was better than 0.99, we saw surprisingly poor open-world performance on the us-west-2 open-world test set by models trained at other vantage points. For the models trained at seven of the nine other vantage points, the FPR at 0.5 recall more than doubled (highlighted in yellow). For three of those, the FPR at 0.5 recall increased by more than an order of magnitude (highlighted in pink). This comparison highlights again the importance of larger-scale open-world testing instead of only closed-world testing. A decline of less than 1% in closed-world accuracy may seem to be an insignificant impact on an adversary's effectiveness, but a 50x higher FPR in a full-scale open-world scenario would have a serious impact due to the base rate problem.

In our Tor experiments, even the closed-world accuracy dropped by as much as 30% when models were trained and tested across some pairs of vantage points, and by as much as 10% on the us-west-2 test set, so the poor generalization across vantage points was more obvious. On the open-world us-west-2 test set, FPRs at 0.5 recall again increased by more than an order of magnitude for models trained at two other vantage points. The FPR increases relative to the baseline were more muted than we saw for HTTPS-only traffic, but the simpler approach (MSP of a deterministic model trained without unmonitored video samples) established a much less effective baseline on the Tor dataset in absolute terms[1], so this does not contradict the worse closed-world results on Tor traffic.

[1]Evidently, open-world Tor fingerprinting benefits much more from the inclusion of unmonitored video training samples.

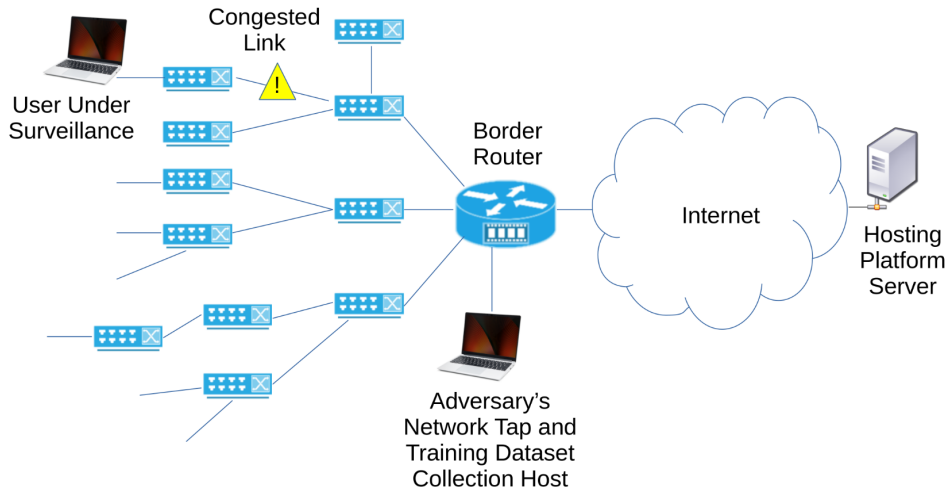


Figure 5: Difference between where a network-level adversary generates the training set and eavesdrops on user traffic, and where users are located on the network. A congested link along some users' path, but downstream from the adversary's vantage point, can cause DASH clients to behave differently at the two different vantage points.

While the results show that employing models across different vantage points can degrade effectiveness, especially in more sensitive large-scale open-world scenarios, we did not find a good explanation for which vantage points produced models that underperformed the baseline and which ones did not. The most we can conclude is that an adversary cannot *assume* that any training data or trained model will generalize well to video traffic streamed from other geographic vantage points. The fact that our vantage points span regions and continents might mean that our results overstate the potential effect of this factor on adversaries that do not have global reach. On the other hand, the fact that our vantage points were all in Amazon Web Services (AWS) datacenters, presumably nearer to the Internet backbone and content delivery network (CDN) servers than most users, might mean that our results understate the potential effect across vantage points with more diverse network conditions.

5.2 Shorter Videos and Traffic Flows

So far, we have limited our analysis to traffic flows of about four minutes in length, corresponding to videos that are four minutes or longer in duration. However, streaming videos on the web may range from a few seconds to a few hours in duration. The literature is broadly in agreement that longer video traffic flows are easier to classify, which is intuitive because – to the extent that all of the features of a longer traffic flow can be represented in a model's input – a model can extract more features to distinguish different videos. While the effects of this on closed-world accuracy have already been shown numerous times, we explore the effect on open-world performance for the first time in this section.

We also hypothesize that the real distribution of video lengths on the web compounds the difficulty of recognizing shorter lengths of video in an open-world scenario. Walsh et al. [62] argued that an adversary who monitors only longer videos by analyzing longer traffic flows also functionally reduces the world size. They noted that Vimeo hosted 650 million videos in 2023, but only about 150 million were four minutes or longer. Therefore, observing a four-minute traffic flow could rule out the 500 million shorter videos as possible matches; the adversary could assume that the flow corresponded to one of the 150 million longer videos. This would mean fewer expected FPs and higher expected precision for any given FPR.

However, if the lengths of videos on the web have a distribution that skews to the short end of the range, it is more likely that videos of interest to an adversary would be shorter. Even if the adversary is primarily interested in longer videos, given the amount of traffic potentially under surveillance and costs of analysis, an adversary might be constrained to analyzing relatively short traffic flows (i.e. only the first few seconds or minutes of longer flows). In those cases, the adversary must contend with the less fingerprintable nature of shorter traffic flows *and* the resulting larger world size, which would compound the difficulty of the task.

5.2.1 Experimental Setup. We first investigated open-world performance as a function of traffic flow length by experimenting with the same training, validation, and testing splits used in our Section 4 experiments. Because the data representation for each traffic flow is the number of bytes sent and received per time step over four minutes, truncating each input to an arbitrarily smaller number of time steps was straightforward. We did this for every multiple of 20 seconds up to the original length of 240 seconds. We added another evaluation at 30 seconds to better define the trend at the short end of the range. We then tuned, trained, and tested fresh models at each length using the baseline approach and one improved approach with Bayesian methods and *mixup* data augmentation that performed well on both the HTTPS-only and Tor test sets in Section 4. We ran 20 trials at each length and calculated the mean for each metric.

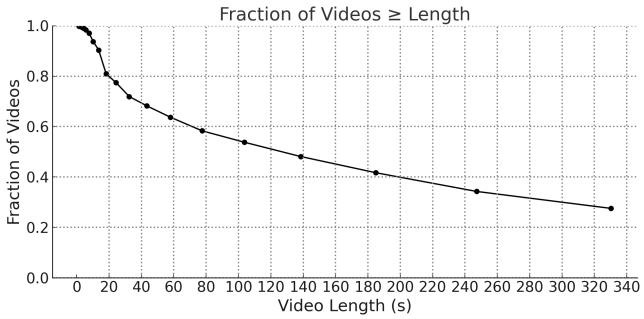


Figure 6: World size as a fraction of all videos, as a function of s, based on the lengths of 10,016 randomly drawn YouTube videos.

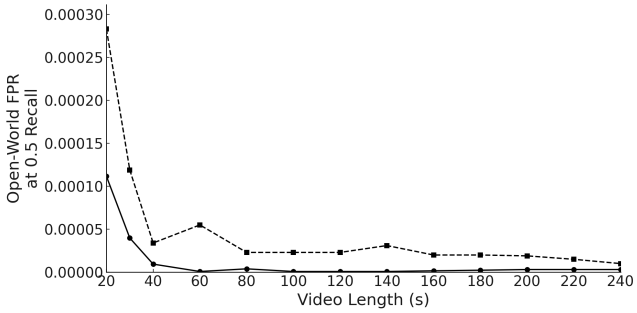


Figure 7: HTTPS-only open-world FPR, at 0.5 recall, as a function of s. The dashed line represents the baseline model performance and the solid line represents the Bayesian model with *mixup*.

We then investigated the distribution of video lengths on the web. Walsh et al. [62] used the total number of Vimeo search results for a set of keywords, when filtered by video duration, to roughly estimate the fraction of videos that were shorter or longer than four minutes. Unfortunately, that method does not allow us to obtain a more granular estimate of the distribution of video lengths. We attempted instead to randomly draw from Vimeo's catalog by scraping search results, but this was difficult due to technical controls that Vimeo recently adopted to combat automated scraping[2]. Fortunately, McGrady et al. [43] were able to randomly draw from YouTube's catalog in 2022 and provide detailed insight into the lengths of uploaded videos. This distribution will of course vary from one hosting platform to another but we believe that YouTube serves as a good proxy for videos on the web in general; it is one of the web's largest video hosting platforms and videos on other hosting platforms are routinely cross-posted to YouTube as well [43].

We transformed the data from McGrady et al. into Figure 6 showing the fraction of all videos that could be a match for a traffic flow of length s (i.e. the world size as a function of the adversary's chosen s). Key to our reasoning is that one cannot assume that users watch the full length of any video, so a traffic flow of length s could match any video of length s or the first s seconds of any longer video, and so the world size grows monotonically as s gets smaller. Reading Figure 6 from right to left, we see that the fraction more than doubled from approximately 35% to 80% as s decreased from 240 to 20 seconds.

5.2.2 Results and Discussion. Figure 7 shows the resulting performance on HTTPS-only traffic flows at each length. We show the mean FPRs at 0.5 recall for the binary open-world task. As we found in the cross-vantage point experiments on the HTTPS-only dataset, open-world testing could reveal significantly worse performance even when closed-world accuracy appeared to remain high. Focusing first on the baseline results: At $s = 40$ with a closed-world accuracy of 0.99, the FPR was approximately 3x higher than it was at $s = 240$. At $s = 30$, FPR was approximately 10x higher even though the closed-world accuracy was still 0.98. Our Bayesian model trained with *mixup* yielded better metrics at every length, but the trend was the same. On the Tor traffic, the metrics were worse at every length, but, again, the trend was the same. There was a strong correlation ($r = 0.992$) between the closed-world error rate (1.0 - accuracy) and open-world FPR, but the open-world FPR allows us to reason about effectiveness at the full-scale of large hosting platforms.

Putting together our understanding of world size as a function of s, and the empirical FPRs as a function of s, we can see the compounding difficulty for an adversary who seeks to monitor shorter videos or limit s due to the costs of analyzing longer flows. The expected number of FPs grows by the product of *two* factors that increase non-linearly as s approaches the minimum length for a given hosting platform. Using Vimeo's estimated size of 650 million videos and our Bayesian model with *mixup* HTTPS-only results at $s = 240$ and $s = 20$ as an example, this number grows from $650,000,000 * 0.35 * 0.000003 \approx 700$ to $650,000,000 * 0.80 * 0.000112 \approx 58,000$; an increase of more than 80x.

6 Limitations and Future Work

The dataset that we used has limitations in terms of both size and realism. While it contains by far the largest number of unique unmonitored videos for open-world analysis, we can already see that our models are pushing the limit of what it can tell us about lower rates of recall. Many individual trials in our experiments resulted in zero FPs at 0.5 and even higher rates of recall, which is partly what necessitated 10-20 trials to obtain useful measurements. Walsh et al.[62] described some of the limitations that are common to synthetic traffic fingerprinting datasets, but we see one more pertaining specifically to our work. It excludes videos hosted by Vimeo that are less than four minutes in duration, while short-form videos might be the fastest growing category of videos on the web. We examined results when truncating the captures down to 20 seconds, but true short-form videos might have characteristics that we could not simulate.

In line with our assumption of the adversary having no prior knowledge, we estimated vulnerable world sizes in Section 4 by assuming that "users select videos to watch in a uniformly random

[2]Vimeo announced new controls in May of 2024 [45] in response to the practices of companies training their large language models. In addition to making future research on fingerprinting more difficult, we believe that the increasing prevalence of techniques to combat scraping also raises the work factor for real-world adversaries.

manner." As we discussed in Section 3, we know that some videos have been exponentially more popular in the wild than others, but view counters by themselves fail to express the fleeting nature of popularity. Estimating more realistic base rates for videos to reason about expected precision remains a major challenge for future work.

There are unanswered questions about what an adversary could accomplish, in terms of increasing effectiveness, through sheer size and computational power. We limited the scope of our work to testing technical enhancements to the baseline approach, building on the same model architecture and collecting no additional training data. There is no obvious limit, however, to the size and complexity of models, nor the quantity of training data, that a nation-state adversary could employ. Future work could experiment with progressively larger models and more training data to reveal a trend, even if academic research cannot feasibly scale beyond some point due to resource constraints.

Given the size and growth rate of most video hosting platforms on the web, we believe it is likely that real-world adversaries must take an open-world approach, and that confronting the base rate problem is their primary challenge. We also found repeatedly that closed-world results can fail to reveal the true difficulty of the task. Accordingly, we call for more work to focus on larger open-world scenarios to continue to gain a better understanding of the effective envelope for fingerprinting.

7 Conclusion

The aim of this study was to better understand the threat of traffic fingerprinting against streaming video on the web. Building on a previously collected dataset and baseline approach, we tested a number of hypothesized improvements for open-world effectiveness. We found that some promising techniques did improve effectiveness, but that attacking users of the largest video hosting platforms would likely still require some prior knowledge to shrink the world size, or would require many observations of a given user over time. Further increasing the adversary's work factor, we found that the effectiveness of our models decreased significantly in two other potentially realistic scenarios: when the training set was not streamed from the same vantage point as the test set, and when analyzing shorter videos or traffic flows. In all cases, we found that Tor still provided a substantial degree of protection against video stream fingerprinting (compared to using an HTTPS-only connection) even under the strong assumption that an adversary can first recognize visits to a certain video hosting platform through website fingerprinting. We do not mean to argue that additional defenses against this threat are unnecessary – that decision must be a function of many more inputs than our findings alone – but we believe that our findings can inform discussion and orient future research on both attacks and defenses in larger and more realistic open-world scenarios.

Acknowledgments

We thank Patrick McClure for his class notes and discussions with us about Bayesian methods for neural networks. We thank David Fifield for his discussions with us about traffic fingerprinting threat models and the importance of studying shorter traffic flows. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

References

[1] Waleed Afandi, Syed MAH Bukhari, Muhammad US Khan, Tahir Maqsood, and Samee U Khan. 2022. A Bucket-Based Data Pre-Processing Method for Encrypted Video Detection. In *Int. Conf. on Comput. Appl. in Industry and Eng.* https://doi.org/10.29007/4rnp

[2] Waleed Afandi, Syed Muhammad Ammar Hassan Bukhari, Muhammad US Khan, Tahir Maqsood, Muhammad AB Fayyaz, Ali R Ansari, and Raheel Nawaz. 2023. Explainable YouTube Video Identification Using Sufficient Input Subsets. *IEEE Access* 11 (Mar. 2023), 33178–33188. https://doi.org/10.1109/ACCESS.2023.3261562

[3] Waleed Afandi, Syed Muhammad Ammar Hassan Bukhari, Muhammad US Khan, Tahir Maqsood, and Samee U Khan. 2022. Fingerprinting Technique for YouTube Videos Identification in Network Traffic. *IEEE Access* 10 (Jul. 2022), 76731–76741. https://doi.org/10.1109/ACCESS.2022.3192458

[4] Sangwook Bae, Mincheol Son, Dongkwan Kim, CheolJun Park, Jiho Lee, Sooel Son, and Yongdae Kim. 2022. Watching the Watchers: Practical Video Identification Attack in LTE Networks. In *USENIX Secur. Symp.* https://www.usenix.org/system/files/sec22-bae.pdf

[5] Alireza Bahramali, Ardavan Bozorgi, and Amir Houmansadr. 2023. Realistic Website Fingerprinting By Augmenting Network Traces. In *ACM SIGSAC Conf. on Comput. and Commun. Secur.* https://doi.org/10.1145/3576915.3616639

[6] Armon Barton. 2018. *Defending Neural Networks Against Adversarial Examples.* Ph. D. Dissertation. Comp. Sci. and Eng. Dept., UTA, Arlington, TX, USA. https://mavmatrix.uta.edu/cgi/viewcontent.cgi?article=1319&context=cse_dissertations

[7] Armon Barton, Timothy Walsh, Mohsen Imani, Jiang Ming, and Matthew Wright. 2025. PredicTor: A Global, Machine Learning Approach to Tor Path Selection. *ACM Trans. on Privacy and Secur.* (Mar. 2025), 1–31. https://doi.org/10.1145/3723356

[8] Abhijit Bendale and Terrance E Boult. 2016. Towards open set deep networks. In *IEEE Conf. on Comput. Vis. and Pattern Recognit.* https://doi.org/10.1109/CVPR.2016.173

[9] Martin Björklund, Marcus Julin, Philip Antonsson, Andreas Stenwreth, Malte Åkvist, Tobias Hjalmarsson, and Romaric Duvignau. 2023. I See What You're Watching on Your Streaming Service: Fast Identification of DASH Encrypted Network Traces. In *IEEE Consum. Commun. and Netw. Conf.* https://doi.org/10.1109/CCNC51644.2023.10060390

[10] Syed M. A. H. Bukhari, Waleed Afandi, Muhammad U. S. Khan, Tahir Maqsood, Muhammad B. Qureshi, Muhammad A. B. Fayyaz, and Raheel Nawaz. 2022. E-Ensemble: A Novel Ensemble Classifier for Encrypted Video Identification. *Electronics* 11, 24 (Dec. 2022). https://doi.org/10.3390/electronics11244076

[11] August Carlson, David Hasselquist, Ethan Witwer, Niklas Johansson, and Niklas Carlsson. 2024. Understanding and Improving Video Fingerprinting Attack Accuracy under Challenging Conditions. In *ACM Workshop on Privacy in the Electron. Society.* https://doi.org/10.1145/3689943.3695045

[12] Giovanni Cherubin, Rob Jansen, and Carmela Troncoso. 2022. Online Website Fingerprinting: Evaluating Website Fingerprinting Attacks on Tor in the Real World. In *USENIX Secur. Symp.* https://www.usenix.org/system/files/sec22-cherubin.pdf

[13] Thilini Dahanayaka, Yasod Ginige, Yi Huang, Guillaume Jourjon, and Suranga Seneviratne. 2023. Robust open-set classification for encrypted traffic fingerprinting. *Comput. Netw.* 236 (Nov. 2023), 109991. https://doi.org/10.1016/j.comnet.2023.109991

[14] Thilini Dahanayaka, Guillaume Jourjon, and Suranga Seneviratne. 2020. Understanding traffic fingerprinting CNNs. In *IEEE Conf. on Local Comput. Netw.* https://doi.org/10.1109/LCN48667.2020.9314785

[15] Ran Dubin, Amit Dvir, Ofir Pele, and Ofer Hadar. 2017. I Know What You Saw Last Minute — Encrypted HTTPS Adaptive Video Streaming Title Classification. *IEEE Trans. on Inf. Forensics and Secur.* 12, 12 (Dec. 2017), 3039–3049. https://doi.org/10.1109/TIFS.2017.2730819

[16] Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a Bayesian approximation: representing model uncertainty in deep learning. In *Int. Conf. on Mach. Learn.* https://doi.org/10.5555/3045390.3045502

[17] Yarin Gal, Jiri Hron, and Alex Kendall. 2017. Concrete Dropout. In *Advances in Neural Inf. Process. Syst.* https://proceedings.neurips.cc/paper_files/paper/2017/file/84ddfb34126fc3a48ee38d7044e87276-Paper.pdf

[18] Zongyuan Ge, Sergey Demyanov, Zetao Chen, and Rahil Garnavi. 2017. Generative OpenMax for multi-class open set classification. In *Brit. Mach. Vis. Conf.* https://bmva-archive.org.uk/bmvc/2017/papers/paper042/paper042.pdf

[19] Chuanxing Geng, Sheng-jun Huang, and Songcan Chen. 2021. Recent advances in open set recognition: A survey. *IEEE Trans. on Pattern Anal. and Mach. Intell.* 43, 10 (Oct. 2021), 3614–3631. https://doi.org/10.1109/TPAMI.2020.2981604

[20] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *Int. Conf. on Learn. Representations.*

https://arxiv.org/pdf/1412.6572
[21] Jiaxi Gu, Jiliang Wang, Zhiwen Yu, and Kele Shen. 2019. Traffic-based side-channel attack in video streaming. *IEEE/ACM Trans. on Netw.* 27, 3 (Jun. 2019), 972–985. https://doi.org/10.1109/TNET.2019.2906568
[22] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Int. Conf. on Mach. Learn.* https://doi.org/10.5555/3305381.3305518
[23] David Hasselquist, Martin Lindblom, and Niklas Carlsson. 2022. Lightweight fingerprint attack and encrypted traffic analysis on news articles. In *Int. Federation for Inf. Proc. Netw. Conf.* https://doi.org/10.23919/IFIPNetworking55013.2022.9829796
[24] David Hasselquist, Ethan Witwer, August Carlson, Niklas Johansson, and Niklas Carlsson. 2024. Raising the Bar: Improved Fingerprinting Attacks and Defenses for Video Streaming Traffic. *Privacy Enhancing Technologies* 2024, 4 (2024), 167–184. https://doi.org/10.56553/popets-2024-0112
[25] Blake Hayden, Timothy Walsh, and Armon Barton. 2024. Defending Against Deep Learning-Based Traffic Fingerprinting Attacks with Adversarial Examples. *ACM Trans. on Privacy and Secur.* 28, 1 (Nov. 2024), 1–23. https://doi.org/10.1145/3698591
[26] Dan Hendrycks and Kevin Gimpel. 2017. A Baseline for Detecting Misclassified and Out-of-Distribution Examples in Neural Networks. In *Int. Conf. on Learn. Representations.* https://arxiv.org/pdf/1610.02136.pdf
[27] Mohsen Imani, Mehrdad Amirabadi, and Matthew Wright. 2019. Modified relay selection and circuit selection for faster Tor. *Inst. of Eng. and Technol. Commun.* 13, 17 (Oct. 2019), 2723–2734. https://doi.org/10.1049/iet-com.2018.5591
[28] Edgar W. Jatho. 2023. *Finding and Fixing Fragility in Machine Learning.* Ph. D. Dissertation. Dept. of Comp. Sci., NPS, Monterey, CA, USA. https://hdl.handle.net/10945/72193
[29] Marc Juarez, Sadia Afroz, Gunes Acar, Claudia Diaz, and Rachel Greenstadt. 2014. A Critical Evaluation of Website Fingerprinting Attacks. In *ACM SIGSAC Conf. on Comput. and Commun. Secur.* https://doi.org/10.1145/2660267.2660368
[30] Chamara Kattadige, Kwon Nung Choi, Achintha Wijesinghe, Arpit Nama, Kanchana Thilakarathna, Suranga Seneviratne, and Guillaume Jourjon. 2021. SETA++: Real-time scalable encrypted traffic analytics in multi-Gbps networks. *IEEE Trans. on Netw. and Service Manage.* 18, 3 (May 2021), 3244–3259. https://doi.org/10.1109/TNSM.2021.3085097
[31] Daniel Kelshaw. 2023. PyTorch implementation of Concrete Dropout. GitHub. https://github.com/danielkelshaw/ConcreteDropout
[32] Muhammad US Khan, Syed MAH Bukhari, Shazir A Khan, and Tahir Maqsood. 2021. ISP Can Identify YouTube Videos that You Just Watched. In *Int. Conf. on Frontiers of Inf. Technol.* https://doi.org/10.1109/FIT53504.2021.00011
[33] Shu Kong and Deva Ramanan. 2021. OpenGAN: Open-Set Recognition via Open Data Generation. In *IEEE/CVF Int. Conf. on Comput. Vis.* https://doi.org/10.1109/ICCV48922.2021.00085
[34] Shu Kong and Deva Ramanan. 2022. OpenGAN: Open-Set Recognition via Open Data Generation. GitHub. https://github.com/aimerykong/OpenGAN
[35] Ranganath Krishnan, Pi Esposito, and Mahesh Subedar. 2024. Bayesian-Torch: Bayesian neural network layers for uncertainty estimation. GitHub. https://github.com/IntelLabs/bayesian-torch
[36] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. In *Advances in Neural Inf. Process. Syst.* https://proceedings.neurips.cc/paper_files/paper/2017/file/9ef2ed4b7fd2c810847ffa5fa85bce38-Paper.pdf
[37] Feng Li, Jae Won Chung, and Mark Claypool. 2018. Silhouette: Identifying YouTube Video Flows from Encrypted Traffic. In *ACM SIGMM Workshop on Netw. Operating Syst. Support for Digit. Audio and Video.* https://doi.org/10.1145/3210445.3210448
[38] Ying Li, Yi Huang, Richard Xu, Suranga Seneviratne, Kanchana Thilakarathna, Adriel Cheng, Darren Webb, and Guillaume Jourjon. 2018. Deep content: Unveiling video streaming content from encrypted Wi-Fi traffic. In *IEEE Int. Symp. on Netw. Comput. and Appl.* https://doi.org/10.1109/NCA.2018.8548317
[39] Shuaili Liu, Licheng Zhang, Peifa Sun, Yingshuo Bao, and Lizhi Peng. 2022. Video traffic identification with a distribution distance-based feature selection. In *IEEE Int. Perform., Comput., and Commun. Conf.* https://doi.org/10.1109/IPCCC55026.2022.9894307
[40] Youting Liu, Shu Li, Chengwei Zhang, Chao Zheng, Yong Sun, and Qingyun Liu. 2020. DOOM: A training-free, real-time video flow identification method for encrypted traffic. In *Int. Conf. on Telecommun.* https://doi.org/10.1109/ICT49546.2020.9239463
[41] Nate Mathews, James K. Holland, Nicholas Hopper, and Matthew Wright. 2024. Laserbeak: Evolving Website Fingerprinting Attacks With Attention and Multi-Channel Feature Representation. *IEEE Trans. on Inf. Forensics and Secur.* 19 (Sep. 2024), 9285–9300. https://doi.org/10.1109/TIFS.2024.3468171
[42] Patrick McClure, Nao Rho, John A Lee, Jakub R Kaczmarzyk, Charles Y Zheng, Satrajit S Ghosh, Dylan M Nielson, Adam G Thomas, Peter Bandettini, and Francisco Pereira. 2019. Knowing what you know in brain segmentation using Bayesian deep neural networks. *Frontiers in Neuroinformatics* 13 (Oct. 2019), 1–12. https://doi.org/10.3389/fninf.2019.00067
[43] Ryan McGrady, Kevin Zheng, Rebecca Curran, Jason Baumgartner, and Ethan Zuckerman. 2023. Dialing for Videos: A Random Sample of YouTube. *J. of Quantitative Description: Digit. Media* 3 (Dec. 2023). https://doi.org/10.51685/jqd.2023.022
[44] Asya Mitseva and Andriy Panchenko. 2024. Stop, Don't Click Here Anymore: Boosting Website Fingerprinting By Considering Sets of Subpages. In *USENIX Secur. Symp.* https://www.usenix.org/system/files/usenixsecurity24-mitseva.pdf
[45] Philip Moyer. 2025. Vimeo's position on AI. Vimeo. https://vimeo.com/blog/post/vimeos-position-on-ai
[46] Lawrence Neal, Matthew Olson, Xiaoli Fern, Weng-Keen Wong, and Fuxin Li. 2018. Open set learning with counterfactual images. In *European Conf. on Comput. Vis.* https://doi.org/10.1007/978-3-030-01231-1_38
[47] Se Eun Oh, Shuai Li, and Nicholas Hopper. 2017. Fingerprinting Past the Front Page: Identifying Keywords in Search Engine Queries over Tor. *Privacy Enhancing Technologies* 2017, 4 (2017), 251–270. https://doi.org/10.1515/popets-2017-0048
[48] Se Eun Oh, Nate Mathews, Mohammad Saidur Rahman, Matthew Wright, and Nicholas Hopper. 2021. GANDaLF: GAN for Data-Limited Fingerprinting. *Privacy Enhancing Technologies* 2021, 2 (2021), 305–322. https://doi.org/10.2478/popets-2021-0029
[49] Se Eun Oh, Nate Mathews, Mohammad Saidur Rahman, Matthew Wright, and Nicholas Hopper. 2021. GANDaLF: GAN for Data-Limited Fingerprinting. GitHub. https://github.com/traffic-analysis/gandalf
[50] Mohammad Saidur Rahman, Nate Matthews, and Matthew Wright. 2019. Poster: Video Fingerprinting in Tor. In *ACM SIGSAC Conf. on Comput. and Commun. Secur.* https://doi.org/10.1145/3319535.3363273
[51] Andrew Reed and Michael Kranch. 2017. Identifying HTTPS-Protected Netflix Videos in Real-Time. In *ACM Conf. on Data and Application Secur. and Privacy.* https://doi.org/10.1145/3029806.3029821
[52] Mohammadreza Salehi, Hossein Mirzaei, Dan Hendrycks, Yixuan Li, Mohammad Hossein Rohban, and Mohammad Sabokrou. 2022. A Unified Survey on Anomaly, Novelty, Open-Set, and Out of-Distribution Detection: Solutions and Future Challenges. *Trans. on Mach. Learn. Res.* (Nov. 2022). https://openreview.net/pdf?id=aRtjVZvbpK
[53] Walter J Scheirer, Anderson de Rezende Rocha, Archana Sapkota, and Terrance E Boult. 2013. Toward open set recognition. *IEEE Trans. on Pattern Anal. and Mach. Intell.* 35, 7 (Jul. 2013), 1757–1772. https://doi.org/10.1109/TPAMI.2012.256
[54] Roei Schuster, Vitaly Shmatikov, and Eran Tromer. 2017. Beauty and the Burst: Remote Identification of Encrypted Video Streams. In *USENIX Secur. Symp.* https://www.usenix.org/system/files/conference/usenixsecurity17/sec17-schuster.pdf
[55] Sandra Siby, Marc Juarez, Claudia Diaz, Narseo Vallina-Rodriguez, and Carmela Troncoso. 2020. Encrypted DNS -> Privacy? A Traffic Analysis Perspective. In *Netw. and Distrib. System Secur. Symp.* https://doi.org/10.14722/ndss.2020.24301
[56] Payap Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. 2018. Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning. In *ACM SIGSAC Conf. on Comput. and Commun. Secur.* https://doi.org/10.1145/3243734.3243768
[57] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. of Mach. Learn. Res.* 15, 1 (Jun. 2014), 1929–1958. https://jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf
[58] Statistics and Data. 2025. Most Visited Websites 1995/2025. https://statisticsanddata.org/data/most-visited-websites-1995-2025/
[59] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2014. Intriguing properties of neural networks. In *Int. Conf. on Learn. Representations.* https://arxiv.org/pdf/1312.6199
[60] Sunil Thulasidasan, Gopinath Chennupati, Jeff A Bilmes, Tanmoy Bhattacharya, and Sarah Michalak. 2019. On Mixup Training: Improved Calibration and Predictive Uncertainty for Deep Neural Networks. In *Advances in Neural Inf. Process. Syst.* https://proceedings.neurips.cc/paper_files/paper/2019/file/36ad8b5f42db492827016448975cc22d-Paper.pdf
[61] Vimeo. 2024. About Vimeo. https://vimeo.com/about
[62] Tim Walsh, Trevor Thomas, and Armon Barton. 2024. Exploring the Capabilities and Limitations of Video Stream Fingerprinting. In *IEEE Secur. and Privacy Workshops.* https://doi.org/10.1109/SPW63631.2024.00008
[63] Tao Wang. 2020. High precision open-world website fingerprinting. In *IEEE Symp. on Secur. and Privacy.* https://doi.org/10.1109/SP40000.2020.00015
[64] Yeming Wen, Paul Vicol, Jimmy Ba, Dustin Tran, and Roger Grosse. 2018. Flipout: Efficient pseudo-independent weight perturbations on mini-batches. In *Int. Conf. on Learn. Representations.* https://openreview.net/pdf?id=rJNpifWAb
[65] Andrew M White, Austin R Matthews, Kevin Z Snow, and Fabian Monrose. 2011. Phonotactic reconstruction of encrypted VoIP conversations: Hookt on fon-iks. In *IEEE Symp. on Secur. and Privacy.* https://doi.org/10.1109/SP.2011.34
[66] Luming Yang, Shaojing Fu, Yuchuan Luo, and Jiangyong Shi. 2020. Markov Probability Fingerprints: A Method for Identifying Encrypted Video Traffic. In *Int. Conf. on Mobility, Sens., and Netw.* https://doi.org/10.1109/MSN50589.2020.00055

[67] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond Empirical Risk Minimization. In *Int. Conf. on Learn. Representations*. https://openreview.net/pdf?id=r1Ddp1-Rb
[68] Licheng Zhang, Shuaili Liu, Qingsheng Yang, Zhongfeng Qu, and Lizhi Peng. 2023. A Novel Adaptive Distribution Distance-Based Feature Selection Method for Video Traffic Identification. In *Adv. Data Mining and Appl.* https://doi.org/10.1007/978-3-031-46674-8_16
[69] Xiyuan Zhang, Gang Xiong, Zhen Li, Chen Yang, Xinjie Lin, Gaopeng Gou, and Binxing Fang. 2024. Traffic spills the beans: A robust video identification attack against YouTube. *Comput. and Secur.* 137, C (Feb. 2024), 103623. https://doi.org/10.1016/j.cose.2023.103623

A Additional Explanation of Bayesian Methods

With dropout [57], for each forward pass during training, for each neuron in a dropout layer, a random variable b_k is drawn from a Bernoulli distribution defined by the dropout rate hyperparameter for that layer, p_{drop}. The activation of a neuron is then set to 0 if b_k is 0.

Spike-and-Slab Dropout [42] combines Gaussian variational inference and dropout. It assumes a variational distribution that is a product of Bernoulli distributions, $Q_{p_{\text{drop}}}(\mathbf{b})$, and Gaussian distributions for each weight, $Q_\phi(\mathbf{w})$. The objective during training is:

$$\underset{\phi}{\operatorname{argmin}}\, KL[q_\phi(\mathbf{b}, \mathbf{w})|p(\mathbf{b}, \mathbf{w}|D)]$$

This reduces to minimizing the negative log likelihood of the data given $\mathbf{b}$ and $\mathbf{w}$, which can be approximated by drawing m Monte Carlo samples of θ from Q and averaging, plus the prior regularization term:

$$\frac{1}{n} KL[q_{p_{\text{drop}}}(\mathbf{b}) q_\phi(\mathbf{w})|p(\mathbf{b})p(\mathbf{w})]$$

The prior regularization term can be rewritten as the sum of the KLD between Bernoulli distributions and the KLD between Gaussian distributions, and there is a closed-form solution for both of these.

In practice, this is mostly a matter of replacing PyTorch Conv1d and Linear layers with BayesianTorch Conv1dFlipout and LinearFlipout layers. The standard forward call transparently performs the Monte Carlo sampling, and we can compute the cross-entropy loss as usual between the outputs and the true labels. A `get_kl_loss()` function returns the sum of the KLD between the Gaussian distributions for every parameter in the model. We only had to write our own function to return the sum of the KLD between the Bernoulli distributions for the dropout layers. We then simply call the standard backpropagation function on the sum of the three losses.

At test time in the context of the Standard Model, a high probability prediction for the $N+1$ unmonitored class and a high probability prediction for a monitored class will both have low entropy (thus low uncertainty). A similar problem arises when using MSP, but the solution when using MSP is to simply exclude the $N+1$ class when taking the maximum. Applying the same solution when calculating entropy, however, does not work because the calculation of entropy assumes a valid probability distribution that sums to 1.0. Our solution was to instead redistribute the predicted probability mass for the $N+1$ class uniformly across the N monitored classes. This procedure has three desirable properties. First, it ensures that the distribution sums to 1.0. Second, it yields maximum entropy when the predicted probability for the $N+1$ class is 1.0. Third, it still yields the minimum entropy of 0.0 when the predicted probability for any monitored class is 1.0. Finally, we made all of the entropy values negative so that the most certain (i.e. lowest entropy) monitored class predictions would have the greatest values, as expected by the scikit-learn `precision_recall_curve()` function.

Table 6: List of geographic vantage points in the dataset.

Location	AWS Region Name
Oregon, United States	us-west-2
Virginia, United States	us-east-1
São Paulo, Brazil	sa-east-1
London, United Kingdom	eu-west-2
Frankfurt, Germany	eu-central-1
Cape Town, South Africa	af-south-1
Stockholm, Sweden	eu-north-1
United Arab Emirates	me-central-1
Seoul, Republic of Korea	ap-northeast-2
Sydney, Australia	ap-southeast-1

B Details of Geographic Vantage Points

Table 6 lists the locations of the AWS regions from which clients streamed videos and collected the traffic in the dataset.

To highlight the differences between geographic vantage points, we analyzed the distances between each vantage point and the Vimeo servers or Tor entry relays to which clients connected to stream video. First, we identified the remote IP address for the heaviest traffic flow in each capture in the dataset. We defined the heaviest flow as having the greatest product of its duration (in seconds between the first and last packet in either direction) and bytes transferred. We then used the MaxMind GeoLite2 City database to resolve IP addresses to countries, cities, and coordinates where possible.

For the Vimeo servers, we were able to resolve 1,687 of 1,908 IP addresses to the city level. In total there were 57 cities. This set of cities intersects with nine of the vantage points from which the clients ran. The median distances to servers ranged from near zero at most vantage points, to 281 kilometers (km) for us-west-2, to 1,150 km for ap-northeast-2. The maximum distances had a much wider range from near zero, at eu-central-1 and eu-north-1, to 12,452 km at ap-southeast-1. We show statistics by vantage point in Figure 8.

We similarly analyzed the distances from the vantage points to the entry relays to which Tor clients connected. Tor clients use weighted random selection, along with some screening criteria, to choose relays in proportion to how much bandwidth the relays provide. This means that the distribution of distances between selected relays should be consistent regardless of client location. However, the distances from clients to entry relays can vary greatly. Because a majority of relays are hosted in Europe, Tor clients elsewhere tend to build significantly longer circuits. We show this in Figure 9. For example, the median distance from eu-central-1 to its selected entry relays was just 892 km compared to 15,867 km for ap-southeast-1. The literature on Tor path selection and performance [7, 27] shows

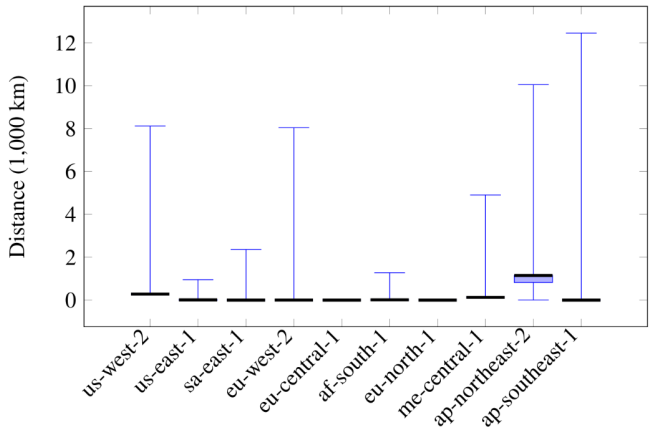


Figure 8: Box plots of geographic distance from vantage points to Vimeo servers in the HTTPS-only dataset. A black line is the median distance from each vantage point.

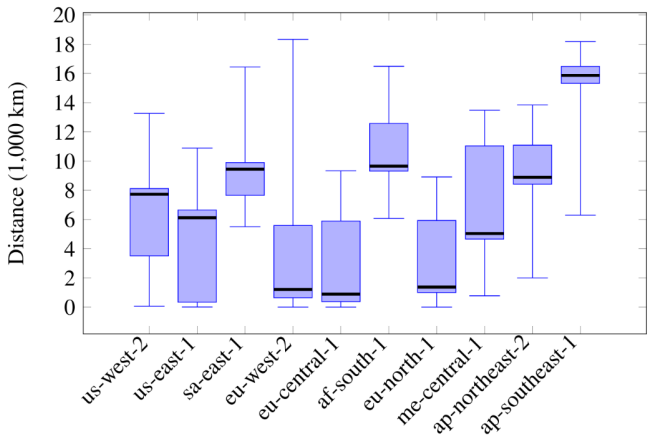


Figure 9: Box plots of geographic distance from vantage points to connected Tor entry relays. A black line is the median distance from each vantage point.

how circuit length can significantly affect the latency and throughput that users experience, so we would expect this to affect the streaming video traffic.