
# NAVAL POSTGRADUATE SCHOOL

**MONTEREY, CALIFORNIA**

# THESIS

**REMOVING THE MASK: VIDEO FINGERPRINTING ATTACKS OVER TOR**

by

Paul H. Duhe' III

March 2023

| | |
|---|---|
| Thesis Advisor: | Armon C. Barton |
| Second Reader: | Gurminder Singh |

THIS PAGE INTENTIONALLY LEFT BLANK

| REPORT DOCUMENTATION PAGE | *Form Approved OMB No. 0704-0188* |
|---|---|



| **1. AGENCY USE ONLY** *(Leave blank)* | **2. REPORT DATE** March 2023 | **3. REPORT TYPE AND DATES COVERED** Master's thesis |
|---|---|---|

| **4. TITLE AND SUBTITLE** REMOVING THE MASK: VIDEO FINGERPRINTING ATTACKS OVER TOR | **5. FUNDING NUMBERS** |
|---|---|
| **6. AUTHOR(S)** Paul H. Duhe' III | |
| **7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)** Naval Postgraduate School Monterey, CA 93943-5000 | **8. PERFORMING ORGANIZATION REPORT NUMBER** |
| **9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)** N/A | **10. SPONSORING / MONITORING AGENCY REPORT NUMBER** |

**11. SUPPLEMENTARY NOTES** The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

| **12a. DISTRIBUTION / AVAILABILITY STATEMENT** Approved for public release. Distribution is unlimited. | **12b. DISTRIBUTION CODE** A |
|---|---|

**13. ABSTRACT (maximum 200 words)**

The Onion Router (Tor) is used by adversaries and warfighters alike to encrypt session information and gain anonymity on the internet. Since its creation in 2002, Tor has gained popularity by terrorist organizations, human traffickers, and illegal drug distributors who wish to use Tor services to mask their identity while engaging in illegal activities. Fingerprinting attacks assist in thwarting these attempts. Website fingerprinting (WF) attacks have been proven successful at linking a user to the website they have viewed over an encrypted Tor connection. With consumer video streaming traffic making up a large majority of internet traffic and sites like YouTube remaining in the top visited sites in the world, it is just as likely that adversaries are using videos to spread misinformation, illegal content, and terrorist propaganda. Video fingerprinting (VF) attacks look to use encrypted network traffic to predict the content of encrypted video sessions in closed- and open-world scenarios. This research builds upon an existing dataset of encrypted video session data and use statistical analysis to train a machine-learning classifier, using deep fingerprinting (DF), to predict videos viewed over Tor. DF is a machine learning technique that relies on the use of convolutional neural networks (CNN) and can be used to conduct VF attacks against Tor. By analyzing the results of these experiments, we can more accurately identify malicious video streaming activity over Tor.

| **14. SUBJECT TERMS** machine learning, video fingerprinting, website fingerprinting, deep learning, convolutional neural networks, attack, defense, adversarial, internet, cyber, dark web, Tor network, The Onion Router | **15. NUMBER OF PAGES** 55 |
|---|---|
| | **16. PRICE CODE** |

| **17. SECURITY CLASSIFICATION OF REPORT** | **18. SECURITY CLASSIFICATION OF THIS PAGE** | **19. SECURITY CLASSIFICATION OF ABSTRACT** | **20. LIMITATION OF ABSTRACT** |
|---|---|---|---|
| Unclassified | Unclassified | Unclassified | UU |

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

**Approved for public release. Distribution is unlimited.**

**REMOVING THE MASK: VIDEO FINGERPRINTING ATTACKS OVER TOR**

Paul H. Duhe' III
Civilian, Scholarship for Service
BA, Bard College, 2018

Submitted in partial fulfillment of the
requirements for the degree of

**MASTER OF SCIENCE IN COMPUTER SCIENCE**

from the

**NAVAL POSTGRADUATE SCHOOL**
**March 2023**

Approved by: Armon C. Barton
Advisor

Gurminder Singh
Second Reader

Gurminder Singh
Chair, Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

# ABSTRACT

The Onion Router (Tor) is used by adversaries and warfighters alike to encrypt session information and gain anonymity on the internet. Since its creation in 2002, Tor has gained popularity by terrorist organizations, human traffickers, and illegal drug distributors who wish to use Tor services to mask their identity while engaging in illegal activities. Fingerprinting attacks assist in thwarting these attempts. Website fingerprinting (WF) attacks have been proven successful at linking a user to the website they have viewed over an encrypted Tor connection. With consumer video streaming traffic making up a large majority of internet traffic and sites like YouTube remaining in the top visited sites in the world, it is just as likely that adversaries are using videos to spread misinformation, illegal content, and terrorist propaganda. Video fingerprinting (VF) attacks look to use encrypted network traffic to predict the content of encrypted video sessions in closed- and open-world scenarios. This research builds upon an existing dataset of encrypted video session data and use statistical analysis to train a machine-learning classifier, using deep fingerprinting (DF), to predict videos viewed over Tor. DF is a machine learning technique that relies on the use of convolutional neural networks (CNN) and can be used to conduct VF attacks against Tor. By analyzing the results of these experiments, we can more accurately identify malicious video streaming activity over Tor.

THIS PAGE INTENTIONALLY LEFT BLANK

# Table of Contents

# List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

# List of Tables

THIS PAGE INTENTIONALLY LEFT BLANK

# Acknowledgments

Thank you to my family, friends and advisors. I owe my success to their love, constant support, and trust.

This material is based upon activities supported by the National Science Foundation under Agreement No 1565443. Any opinions, findings, and conclusions or recommendations expressed are those of the author and do not necessarily reflect the views of the National Science Foundation.

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 1:
# Introduction

## 1.1 Introduction

In the current state of society, many people, regardless of age, location or profession find themselves using the Internet in a personal and/or professional capacity. For many, online privacy and anonymity are major concerns. Online privacy can be defined as the ability to hide or obscure ones actions while using the internet or online services. Anonymity makes the distinction in that, while a user's actions may be seen, their identity is kept hidden. While conceptually simple, anonymity and privacy on the internet is not easily obtained. Due to the obvious need, many software companies have attempted to develop and construct infrastructures for facilitating anonymous and private communications. Virtual Private Network (VPNs), proxy servers, and SSH tunneling are just a few methods in which users can attempt to hide their identity and actions while using the internet. This research focuses on The Onion Router (Tor). Tor, created by the non-profit organization the Tor Project, is a free and open-source, low-latency, privacy-preserving framework used by millions of users worldwide [1]. Using a distributed overlay network of over seven thousand volunteer relays, Tor routes users' traffic in order to obscure and make it difficult to trace both the source and destination IPs used in a given session[2].

As basic online privacy rights continue to be placed under judicial scrutiny, open-source tools for preserving anonymity, like Tor, are increasingly important. Services that provide users the power to easily encrypt their network traffic and online session data is a necessity that allows users to feel safe when sending sensitive data over the internet. Unfortunately, this is a power that is realized and exploited by threat actors.

While traditional web browsing can be done using the Tor Browser, Tor is more commonly known as a tool for accessing the Dark Web. The internet is often described as consisting of multiple layers. The surface layer, which consists of websites like YouTube, Twitter, and Facebook, can be accessed through traditional internet browsers using a website's distinct uniform resource locator (URL) [3]. In contrast, the deepest layers of the internet, infamously

known as the Dark Web, can only be accessed through specialized software such as the Tor Browser.

The Dark Web has a history of hosting sites used for selling and distributing drugs, child pornography and firearms. Historically, terrorist organizations have successfully leveraged the Dark Web to organize, recruit and carry out attacks. Tor's encryption and privacy-preserving capabilities have made it an easy choice for Salafi-jihadist violent extremist organizations (VEOs) and the Islamic State (IS) to host online message boards, recruitment propaganda, and facilitate secure communications amongst members [4]. Examples include the Isdarat, established in 2015, which "constituted the most abundant repository of IS propaganda on the dark web" [4] as well as the Shumukh al-Islam message board and Kavkaz Center (KC).

Tor was believed to be used by ISIS to organize and communicate before and after the 2015 Paris attack, which led to over 130 deaths and more than 400 injured French citizens. In response, France attempted to ban the use of the Tor Network during states of emergency [5] to more successfully track perpetrators.

These usages show that Tor has the capability to be used for malicious means, causing at worst the loss of life. These malicious uses beg the question, "Is encrypted traffic over Tor completely unbreakable?"

Despite its aforementioned complexity, Tor is known to be vulnerable to traffic analysis [6], an attack on a user's privacy. Modern websites transfer large amounts of data and respond dynamically to user inputs and keystrokes. It has been shown that website identification is possible based on variations in packet size and packet timing as a result of these interactions [7].

Website Fingerprinting (WF) is a traffic analysis attack that allows an attacker to capitalize on these differences in network traffic based on variations in website content. Using machine learning (ML), an adversary will train a classifier using quantifiable network traffic features such as packet time, packet direction, and packet size [8]. When successful, the attacker should be able to determine the website visited by the victim.

Similar to website fingerprinting, a video fingerprinting (VF) attack allows an attacker to

use variations in encrypted network traffic to determine the video being watched by a Tor user. As the modern internet moves to abundantly consist of video streaming traffic, it can be predicted that terrorist organizations will use videos to communicate, provide instructional material, and spread terrorist propaganda. This research aims to delve further into the field of video fingerprinting to potentially thwart terrorist organizations who wish to use Tor to spread malicious content in video format.

## 1.2 Purpose and Scope

The aim of this research is to use Deep Fingerprinting techniques proposed by Sirinam et al. [6] for website fingerprinting to train a video fingerprinting classifier to accurately identify videos watched over an encrypted Tor connection. By adding to a pre-existing dataset of network features extracted from encrypted video streaming sessions, this research will investigate the use of Deep Learning (DL) techniques to classify specific videos and video genres. Deep Learning has been "shown to outperform traditional machine learning techniques in many domains, such as speech recognition, visual object recognition and object detection [6], [9]. Furthermore, "DL does not require selecting and fine-tuning features by hand" [6], [10].

Our novel contributions include adding an additional genre to the dataset of video traffic traces to potentially increase accuracy and minimize overfitting during genre classification. Additional data should also help provide more accurate results, given a more realistic dataset size in regards to video fingerprinting attacks. Further, we will explore more scenarios that could help identify features that could positively affect the accuracy results of our tests. These scenarios include smaller traffic trace sizes and the use of packet size data in genre and video classification. We will also explore the use of open-world machine learning to mimic an attack scenario where the attacker does not have access to all possible videos watched by a victim. Using a monitored and unmonitored dataset we will attempt to classify videos based on genre in an open-world scenario.

## 1.3 Thesis Organization

The remaining chapters are organized as follows. Chapter II will provide an overview of Tor network connections, the Dark Web and cybercrime, as well as the machine learning

methods used in this research. Chapter III will discuss the tools used to capture the Tor traffic for our dataset and method for training our classifier. Chapter IV will delve into the research itself and discuss findings. Finally, Chapter V will cover ideas for future research and any additional thoughts on the state of research in the field of video fingerprinting.

# CHAPTER 2:
# Background

The invention of the internet and other methods of computerized communication have drastically improved multiple facets of the human experience and increased economic, military and scientific capabilities across nations[11]. Along with these advancements came a host of unknown malicious possibilities supported by the infrastructure of the internet. Since the beginnings of the internet, originating with ARPANET (Advanced Research Projects Agency Network) developed by the Defense Advanced Research Projects Agency (DARPA)[12],computer viruses and malware have been commonplace.

In 1971, the first computer virus, the "Creeper Virus", with the ability to replicate itself across a network, was released. [11] For the following 50 years, malicious computer programs would continue to become more sophisticated and consequential to individual users, large corporations and nation states. Cyber crime and malicious attacks in the form of malware have become an ever-growing norm. Due to this ever growing threat, the Department of Defense (DoD) has recognized offensive and defensive cyber security as a priority. Adversaries of the United States, including China and Russia, are utilizing cyber espionage in their military, political and economic endeavors [13].

The internet has provided radical groups with a medium from which to communicate, spread ideas, and recruit new individuals to their organizations. Threat actors with various intentions and motives launch attacks on small businesses, large corporations and nation states across the globe.

Cybercrime is a fast growing and profitable industry. With a projected 13.1 billion IoT devices connected to the internet in 2022 [14], cybercrime has the unique ability of affecting hundreds of millions of people. Perpetrators also have many resources at their disposal to preserve their anonymity, making cybercrime a low cost- high reward endeavor. No tool has been as influential as the Onion Router (Tor) in assisting threat actors in preserving their anonymity while online.

## 2.1 The Tor Network

Tor, created by the non-profit organization the Tor Project, is a free and open source privacy framework used by millions of users worldwide. Through the use of an overlay network consisting of over seven thousand volunteer, relay nodes, Tor routes users' traffic multiple times to obscure the location and intended destination of network packets.

The Tor overlay network consists of three types of nodes: Onion Routers, Onion Proxies, and directory authorities. [15] Onion Proxies refer to the proxy initialized and maintained by a local Tor user, typically done through the use of the Tor Browser. Onion Routers are volunteer run relays that are used to send network traffic over the Tor network. During normal operation, a client's onion proxy will contact a directory authority to retrieve the network consensus, which is "a single document compiled and voted on by the directory authorities once per hour, ensuring that all clients have the same information about the relays that make up the Tor network." [16]

The Onion proxy is then responsible for constructing a Tor circuit consisting of onion routers. The circuit is constructed by selecting three relays, factoring each node's bandwidth capacity. Traffic is then encrypted and split into 512 fixed byte cells to send across the constructed Tor circuit.

When connecting to the Tor Network, a user's data is routed through various types of volunteer nodes (onion routers) before finally connecting to the internet. These nodes can be classified into four categories: **Guard nodes**, **Middle nodes**, **Exit nodes**, and **Bridge nodes**. When a user connects to the Tor network, a "circuit" is built that typically consists of a guard or bridge node, a middle node, and an exit node. This three-hop circuit is what defines a typical Tor connection and what Tor's decentralized anonymity is based on. A special Tor relay called a direct authority is used to maintain a list of currently-running and accessible relays[16].

**Tor guard nodes** mark the point of entry for a client into the Tor network. Being the first node in a Tor circuit, Guard nodes sit in a relatively powerful position as they can observe the IP of the connecting user. Because of this, Guard nodes must meet several requirements and go through a rigorous vetting process before they are allowed to act as a guard node. Some of these requirements relate to the length of time the current node has been part of the

network, as well as bandwidth requirements to ensure relatively fast access to the network. Guard nodes must be stable and fast (at least 2MByte/s).

As noted in [17] a guard node is required to have the appropriate flags to be accepted as a guard node. Specifically, the "guard-wfu" flag and "guard-tl" flag denote that a specific Tor relay is a guard node. Because these nodes act as the gateway into the Tor network they have the privileged vantage point of viewing the IPs of users connecting to Tor. A compromised guard node could completely unravel the trust expected from the Tor network, which explains why more intense specifications are needed to operate as a guard.

A Tor user generally will connect to the same guard node for up to nine months before they are assigned a new guard or their current guard becomes unavailable [18].

**Middle nodes** exist between guard and exit nodes.

**Tor exit nodes** are the last node that a client's traffic will pass through before establishing an internet connection. Similar to Guard nodes, Exit nodes sit in a privileged position. The Exit node is responsible for shedding the last layer of encryption before delivering the traffic to its intended destination on the public internet. For this reason, Exit nodes typically are more vulnerable to subpoenas, legal action, and abuse complaints as any illegal network traffic leaving the Tor network appears to be coming from the Exit node's IP. Further, a malicious Exit node operator can effectively sniff the traffic coming through Tor and analyze it unencrypted. The Tor Project provides information and response templates for operators in case of abuse complaints.

**Tor Bridge nodes** are a special type of node that are primarily used to help bypass censorship and blocked websites in countries where censorship is the norm. Bridge nodes accomplish this by not providing public lists of bridge nodes in direct authorities. Instead, a bridge authority maintains a list of bridge nodes. Bridge authorities attempt to protect against node enumeration using various tools and implementing certain security features. This helps prevent the public release of all bridge node IPs.

These volunteer nodes are what provides Tor with its decentralization while also being the crux of one of its main performance issues. Disinterest in TOR typically comes from its slow performance, which can be partially faulted to low bandwidth relays[15]. This problem

is exacerbated with the use of bridge nodes.

### 2.1.1 Tor Encryption

To protect a client's data as it is traversing a Tor circuit, onion routing is utilized to encrypt the original message of the user. Onion routing was originally designed in the 1990s at the US. Naval Research Laboratory. Using asymmetric encryption the Tor service encapsulates a user's "message" in multiple layers of encryption. The keys for each of these layers is only known by a singular node in a Tor circuit. Further, when decrypting or unravelling a layer of an onion's encryption, all that is seen by the intermediary node is the source and destination of the current packet. This protects the content of the current message as well as the origin and final destination of the message.

## 2.2 Tor Vulnerabilities and Attacks

As previously stated, Tor Guard and Exit nodes exist in privileged positions in a typical Tor session. An adversary with the ability to control a portion of these nodes could use this ability to de-anonymize users, inject malicious data, and conduct man-in-the-middle attacks like website fingerprinting. This type of attack is called a Sybil Attack. Sybil Attacks against Tor define a scenario where some global adversary with the appropriate resources could create a large number of nodes, disproportionately influencing the Tor network [19]. Coined by Douceur [20], and named after a woman suffering from dissociative identity disorder from the 1973 novel Sybil, Sybil attacks are challenging to discover and to defend against.

As further stated in "Tor: The Second-Generation Onion Router", written by the founders of Tor, "A global passive adversary is the most commonly assumed threat when analyzing theoretical anonymity designs. But like all practical low-latency systems, Tor does not protect against such a strong adversary[21]."

The effectiveness of Sybil Attacks grow as the proportion of nodes controlled by an adversary increases. With a relatively large percentage of control over the Tor network the following attacks become possible [19]:

- **Exit traffic tampering:** With control of an exit node, an adversary has the ability

to tamper with traffic as it leaves the Tor network. This could lead to simple traffic sniffing to the more advanced modification of the traffic leaving the network. This was witnessed in the 2020 Bitcoin Address Rewrite Attacks that led to attackers modifying the destination of Bitcoin transactions and routing Bitcoin to their own accounts.

- **Website Fingerprinting:** With access to encrypted network traffic (that could be captured at any point in a Tor circuit), machine learning can be used to use distinct traffic features to predict the sites visited by a Tor user.
- **End-to-end correlation:** In a much larger scale attack, if an adversary controlled both entry and exit nodes in a Tor circuit, they could use a timing attack to completely deanonymize the Tor user and reveal the traffic and intended destination of the traffic.
- **Bridge address harvesting:** With control of middle nodes, adversaries can harvest the IPs of all source nodes and correlate this with all publicly known Guard nodes. Any node IPs found that aren't found in the public registry of Tor nodes would reveal the IPs of bridge nodes, deanonymizing users who are connecting to Tor from countries that are looking to censor their citizens, and reduce privacy.

Because guard nodes and exit nodes are easily distinguished as Tor related, many services and ISPs will knowingly block connections from these IPs. Tor bridge nodes attempt to circumvent this issue. Bridge nodes are hidden guard nodes that are low bandwidth but allow users with more restricted internet privileges to connect to Tor.

Tor is also weak to end to end attacks, where some entity has control over both the guard node and exit node. Without a large number of users it can be easy to identify a Tor user. In one scenario authorities could see the exit node IP was a Tor node, and because only one host was accessing Tor at the time on the network (connecting to a Tor guard node) they were able to easily identify the perpetrator.

Tor is a powerful tool for encrypting a users' traffic, however it is not as effective at hiding a user's identity. With information from a user's ISP, network admin and such, it can be deduced where the Tor traffic originated from. The content will be encrypted, however other features, like the amount and size of the data being transferred will also be visible. This is where fingerprinting shines and begins to breakdown the anonymity provided by Tor.

## 2.3 Website Fingerprinting

Website Fingerprinting (WF) is an adversarial, man-in-the-middle attack that poses a threat to the confidentiality and anonymity provided by Tor. The attacker in a WF attack is assumed to be a passive, network level entity who can intercept a Tor user's encrypted network traffic. Here, *passive* denotes that the network traffic is *only* intercepted by the attacker. The attacker in this context does not have the ability to modify, drop, or add packets to the user's traffic [22]. A WF attack also assumes that the traffic intercepted is encrypted, as with an unencrypted session, fingerprinting would not be necessary.

By utilizing traffic direction and burst-level timing features, machine learning (specifically deep learning) has been proven successful at classifying websites visited by the victim [22]. A WF attack utilizes differences found in website content and network traffic characteristics to identify these websites. Previous research has found "timing features" and specifically "burst-level timing features" extremely accurate (up to 98% accuracy [6]) at predicting website content.

Website Fingerprinting is typically a classification problem, utilizing machine learning to train a classifier to predict websites against a control set. First, the researcher will collect network traffic by visiting a set of previously selected websites. Then, with selected features, including packet times, length, and direction, training will begin against the collected data [8]. Each captured network trace represents a simulated occurrence of the victim visiting the chosen website. The *network-level* attacker is assumed to be able to capture the victim's traffic as if connected to the victim's network locally, by controlling a guard or middle node in the victim's Tor circuit, or by controlling a router of the victim's ISP. It is assumed that the traffic collected and analyzed is encrypted in some form, ie. through the use of a service like Tor.

**Closed World** analysis assumes that a $w$ number of websites exist in the world and users can only visit websites from this set [23]. It is generally assumed that the number of websites included in this group is an extremely small percentage of sites, compared to the number of actual websites in the world. This method is less realistic, however is useful as a test case for the attacker's machine learning classifier.

**Open World** analysis still uses a small set of websites as a control group, but does not limit

the victim/user to these sites for visiting. This control group of chosen data points is called the *monitored set*. All other websites in the open world would form the *unmonitored set*. An open world classifier is then used to predict websites visited from the entire world of websites possible to visit.

### 2.3.1 Previous Work

Many attacks have been developed by researchers using ML to bypass encryption implemented by privacy-preserving software (such as Tor). Varying techniques were used and developed each with a range of results.

Herrmann et al. [24] of the first to implement closed-world WF methods to conduct attacks against Tor. They developed a WF technique based on a Multinomial Naïve-Bayes (MNB) classifier[24]. However, while this method was successful against multiple forms of encryption, their MNB classifier was only 3% accurate against Tor, using a monitored set of 775 sites. Sirinam et al. [6] identifies shortcoming of this research as a "reliance on packet length frequencies". The development of more dominant and distinctive features would eventually improve the accuracy achieved with similar WF methods.

Wang et al. proposed the use of a k-nearest neighbors ($k$-NN) classifier attack to defeat common defenses against WF attacks [25]. They found that despite using a large database of websites, the attack was relatively fast (matter of seconds) compared to other methods. Further, with a correct selection of features and $k$ can perform well in open-world scenarios. "In a closed-world setting with 100 sites, it achieved 91% accuracy, and in an open-world setting with 5,000 sites, it achieved 86% True Positive Rate (TPR) and 0.6% False Positive Rate (FPR). [6]"

Sizable research has also been done to defend against traffic analysis attacks against Tor. In Sirinam et al. [6] Deep Fingerprinting research, fingerprinting attacks were conducted against defended and non-defended Tor traffic. They utilized models to classify traffic against WTF-PAD [26] and WalkieTalkie defenses [27]. These models are discussed briefly below. Since this research conducted in 2018, more novel traffic defenses have been proposed and implemented by Tor including Mockingbird [28], Dolos [29] and TrafficSilver [30]. As our data collection practices center around collecting Tor traffic, this research will focus on classifying traces from modern Tor defense mechanisms implemented in the Tor Browser

Bundle version 8.0.2. However, it is worth mentioning for future work collecting defended Tor traffic and testing defenses against this traffic.

**WTF-PAD (Website Traffic Fingerprinting Protection with Adaptive Defense)** is an adaptive padding (AP) technique that has been proven successful at defending against WF attacks [26]. Originally developed by Shmatikov and Wang to combat end-to-end traffic analysis [31]. Juarez et al. [26] were able to adapt their research into an implementation appropriate for defending against WF attacks on Tor. In their original research, Shmatikov and Wang recognized the importance of *bursts* within network traffic over Tor (and what would come to be important in research regarding VF). In response, they configured their AP technique to send bursts of traffic within large gaps of legitimate traffic [26], [31]. Juarez et al. [26] developed methods for better disguising fake packets from legitimate ones and configuring their defense to better mimic legitimate behavior. They used a *bootstrapping* phase to collect data on expected traffic parameters (burst, gap, and inter-arrival times) and used this to send dummy packets in traffic gaps, accurate to the degree of milliseconds [26].

**WALKIE-TALKIE** is a WF defense developed by Wang and Goldberg. WALKIE-TALKIE modifies the typical user's internet browser to use half-duplex communications vs the typical full-duplex communications [27]. Their proposed half-duplex mode allows for easier modification of burst sequences found within encrypted network traffic with limited overhead. Wang and Goldberg claim that WALKIE-TALKIE is effective against WF attacks against Tor as well as other methods [27]. Sirinam et al. [6] Deep Fingerprinting and show very high accuracy scores when testing with no defenses and against WTF-PAD. However, Deep Fingerprinting remains ineffective against WALKIE-TALKIE [6].

### 2.3.2 WF Limitations

While previous fingerprinting work has been successful and resulted in high accuracy when classifying websites against a given set of websites, there are limitations to the approaches tested [8]:

- **Tor Browser Bundle Differences:** Previous research assumes similar versions of the Tor Browser Bundle (TBB) used between the attacker and user.
- **Lack of Website Variation:** Localized copies of websites used, assume users will be visiting the exact same copies of websites with no variations between visits.

- **Multi-tab Browsing Sessions:** WF studies done assume a single-tab session when simulating these attacks. However, real world browsing sessions most likely would include users who have multiple tabs open at a time. This could affect the efficacy of these studies and results found. This research does provide a upper bound for the capabilities of website fingerprinting attacks given the near perfect conditions tested.

## 2.4 Video Fingerprinting

Similar to WF, Video Fingerprinting (VF) is a machine learning classification problem used to make predictions on a victim's encrypted browsing session. Schuster et al. [32] discusses encrypted video streams and identifies characteristic patterns in encrypted TCP streams of video streaming traffic. While strong encryption may be able to hide the content of network traffic, it is less successful at disguising network traffic patterns such as bits transmitted per second. Schuster et al. [32] describes the MPEG-DASH streaming video standard and describes the patterns exhibited by this traffic as *bursty*. "The MPEG-DASH streaming standard (1) creates video segments whose size varies due to variable-rate encoding, and (2) prescribes that clients request content at segment granularity. [32]" Bursts can be defined as consecutive groups of network packets sent from client to server or vice-versa [22].

THIS PAGE INTENTIONALLY LEFT BLANK

# CHAPTER 3:
# Methodology

The initial stages of this research were dedicated to studying previous work in this field, including a background literature review and an analysis of the machine learning tools needed to accomplish related tasks. Following a literature review related to Tor, Website Fingerprinting and Video Fingerprinting, the research moved into the phases related to building and testing our own VF efforts. This chapter consists of steps for data collection and machine learning.

## 3.1 Data Collection

The process of collecting data to add to the video dataset marked the initial phase of the research portion of this project. This phase consisted of choosing a dataset for collecting encrypted video traffic, selecting a site for collecting the data and finally the use of the Tor Browser Crawler, a tool used for scraping encrypted network traffic data from Tor.

YouTube.com was selected as the website for scraping data. YouTube is currently the most popular video streaming site, and second only to Google as the most visited website in 2021 and 2022 [33], [34].

As this research is a continuation of work done by [35], [36], the dataset used is a combination of data collected during those research projects and the addition of one genre, added during this research.

During research conducted by Campuzano et. al [35], a set of videos were chosen for his research. The videos were all related to each other and formed a single genre of videos. For his research Campuzano used YouTube as the platform from which to scrape network traffic. The dataset used for Campuzano's research consisted of nine popular Disney movie trailers.

Following Campuzano's research, Kim et. al, [36] extended this dataset by adding 2 additional genres. The focus of her work broadened the classification possibilities. Instead of

classifying videos based on Video ID, Kim extended the classifier to classify videos based on their content or more specifically genre. Kim's additions to the research included the aforementioned 2 additional genres and results from her testing. The added genre's included popular music videos and professional sport footage.

As this research assumes a scenario where terrorist organizations are using video streams as the format for spreading propaganda and illegal content, the genre added in this portion of research consists of footage containing explosions and the use of bombs and other explosives. Table 3.1 includes genre 3 video names, the video lengths, and video ID of the videos in the dataset.

Table 3.1. Genre 4 - Explosion Footage Dataset

| Video ID | Video Name | Video Length |
|---|---|---|
| 28 | Taliban releases a drone footage of suicide bombing | 0:49 |
| 29 | Twin suicide bombing kills at least 28 in Baghdad market | 1:43 |
| 30 | ISIS launches multiple suicide attacks in Baghdad | 2:09 |
| 31 | 8 Different angles of the Boeing 767 hitting the south tower of the World Trade Centre | 1:17 |
| 32 | Boston Marathon explosions video: Two bombs near finish line | 2:52 |
| 33 | Barrel bombing campaign intensifies in Aleppo, Syria | 1:59 |
| 34 | Video captures bombs exploding in Syrian City of Daraya | 0:49 |
| 35 | Syrian rebels release footage of a bomb attack | 0:43 |
| 36 | Moment of Kabul bombing captured by a security cam | 0:44 |

## 3.2 Tor Browser Crawler

Collecting and building a dataset for analysis was a large component of this research project. Gathering data that would allow for sound machine learning practices to be used was crucial. To remain consistent with previous work by Kim and Campuzano, the *tor-browser-crawler* was used for collecting data [35], [36]. This tool mimics a typical Tor user browsing session, and assumes an adversary who can access data between the victim user and the victim's Tor entry node. [6]

The Tor Browser Crawler is written in Python and utilizes *tcpdump* to capture network traffic. *Tcpdump* is a command line packet analyzer that can be used to capture network traffic over a network interface. The *tor-browser-crawler* initializes *tcpdump* with flags to

specify the output file to write captured packets (-w), the interface from which to capture packets (-i), and the number of seconds before the dump file is rotated (-G).

The crawler uses a list of URLs saved in a file named "videos.txt" to specify the sites to be visited by the crawler. The crawler then navigates to each video sequentially, during which a PCAP file is created containing captured packets from each successful crawl, and screenshots are captured showing the state of the site during the crawl. These PCAPs are saved in separate directories for each iteration through the videos.txt file.

## 3.3 PCAP Parser

After completing the data collection phase, a PCAP parser was used to extract the important features from the packet captures. The parser, which was written in python, uses the *scapy* library to parse the collected packet captures. *Scapy* is a python module that can parse, decode, and capture network traffic. *Scapy* was used in this context to read each packet capture, and pull packet times, packet directions, and packet sizes from each trace. These features were saved into a text file for each crawl displaying these features in a list, beginning from the earliest capture packet to the last packet of the capture. These parsed PCAPs are then fed into our machine learning algorithm to look for patterns in the encrypted traffic.

## 3.4 Threat Model

Our model assumes a passive and local adversary who has the ability to intercept and eavesdrop on a user's encrypted Tor session. The attacker may achieve this by gaining access to their victim's WiFi network, or compromising the victim's ISP. While the attacker can view the encrypted traffic our model assumes they cannot add, drop, or modify packets. Further, decrypting the traffic is also not an option.

Juarez et. al discusses two styles of attacks [8]: a targeted and non-targeted attack. During a targeted attack an attacker targets a single victim and is able to eavesdrop and collect their browsing data. The attacker can train a classifier on the collected data to attempt to fingerprint. This style of attack has to potential of being more successful than the latter as the attacker can use information gathered about the specific victim to more accurately train their classifier. A non-targeted attack is an attack against many users. This is often done

from the vantage of an ISP, entry guard operator or Internet Exchange. As these entities have the privilege of seeing traffic from many users they can conduct attacks against traffic from many users.

Large assumptions about the victim's browsing behavior are made in our model. As the Tor Crawler visits one site at a time, the assumption is that a user has only one active session and only one tab open at a time. The user is expected to view videos sequentially, watching the video in its entirety before moving to the next. While this may not reflect normal browsing behavior of internet users it may be more accurate of Tor users since Tor's speed limits the intensity of a user's browsing sessions.

## 3.5 Lab Environment

Two separate environments were used for this research. The first was designed for collecting network traffic traces over the Tor network, and the second for training our machine learning model using the collected data.

**Crawler Environment:** To accurately simulate a typical Tor users' environment, we chose to run our experiment within an environment that closely resembled that of a normal user. Our lab environment was therefore run from a home network that closely resembled the network speed and bandwidth expected of a normal home user. The crawler was run on an install of Ubuntu running on an Intel NUC.

**Training Environment:** This research utilized the Naval Postgraduate School's (NPS) Hamming supercomputer [37] for data processing and running our machine learning models. This architecture allows for the use of NVIDIA GPUs which increased processing speed and decreased training times by a significant amount.

# CHAPTER 4:
# Test Design and Implementation

This chapter will discuss the structure of the data used for our VF classifier as well as scenarios chosen to simulate various attack scenarios. For each scenario we will discuss the reasoning, implementation and results achieved. Lastly, we will examine real world implications based on the observed results.

## 4.1 Feature Selection

After collecting and parsing all network traffic traces using tor-browser-crawler and pcap parser, the dataset was read into a pandas dataframe. The dataframe represented the captured data with the following features (stored as df columns):

- **Genre:** Genre classification labeled 0-3. Genre 0: Disney Movie Trailers, Genre 1: Music Videos, Genre 2: Sports Clips, Genre 3: Explosion Footage.
- **Video ID:** Unique Video Id for each video in dataset, labeled 0-35 to include all 36 videos included in research:
  - **Genre 0:** Video Id 0-8
  - **Genre 1:** Video Id 9-17
  - **Genre 2:** Video Id 18-26
  - **Genre 3:** Video Id 27-35
- **Packet Times:** An array of times, imposed on the dataframe as a unique column representing each possible time entry. This metric is padded with zeros to ensure uniformity between traces.
- **Packet Directions:** An array of packet size and directions imposed on the dataframe as a column for each entry. A positive number represents an outgoing packet, while a negative number represents an incoming packet.

## 4.2 Training Scenarios

To compare accuracy results with those observed by Kim and Campuzano [35], [36] multiple tests were run using the larger dataset containing the additional fourth genre (explosion

footage). While Tor employs encryption and protections on its traffic, these protections are susceptible to traffic analysis attacks. Lightweight defense mechanisms have been proposed to help circumvent traffic analysis attacks including MockingBird [28], TrafficSliver [30], WTFPAD [26], Walkie-Talkie [27] and others. Our data collection efforts focus on defense mechanisms inherently implemented by Tor Browser Bundle version 8.0.2 and nothing else. For this reason we will refer to our dataset as non-defended. Similar to research conducted in [35] [36], our dataset is collected with no defenses added. However, in this previous research these non-defended sets were tested against classifiers used for defense models WalkieTalkie and WTFPAD. As this does not provide relevant results considering the traffic does not implement these defenses, our research will solely focus on testing non-defended traffic using classifiers expecting non-defended traffic. Figure 4.1 shows a comparison of accuracy results for genre and video ID prediction between Campuzano [35], Kim [36] and this research.

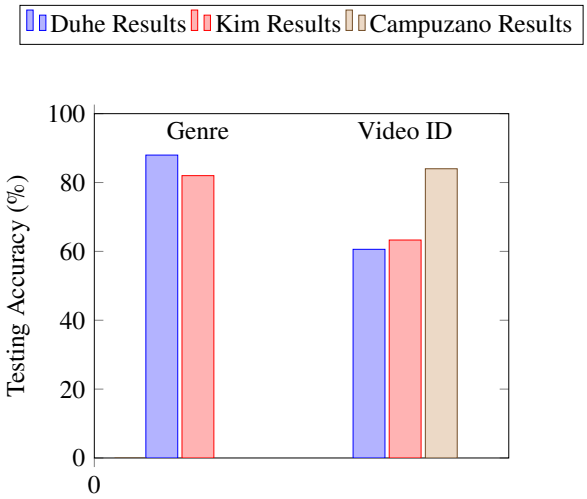


Figure 4.1. Accuracy Results (packet size removed) between Kim [36], Campuzano[35] and Duhe

**Video ID Classification:** Campuzano's original dataset consisted of one genre containing nine videos. Each video was given a unique Video ID ranging from 0-8. Campuzano observed an 84% accuracy when training with a non-defended dataset classifying captures based on video ID. Kim experienced lower accuracy scores when classifying based on Video ID, and mentions that this is likely due to the increase in number of classes used in her research. Campuzano had nine classes whereas Kim's dataset consisted of 27 classes.

Kim observed a 63% accuracy when classifying videos based on video ID, with a dataset of 27 videos. To compare results, training will be done using a larger dataset, now consisting of 36 unique videos. Each video will be given a unique video ID ranging from 0-35, and accuracy will be compared to that observed during previous research.

**Genre Classification:** Kim's research expanded the dataset by two additional genres (18 additional videos) allow "ing for classification to be done on video genre as opposed to only video ID. Each video was labeled with a genre ranging from 0-2. The genres included Disney Movie Trailers, Sports Clips, and Music Videos. Kim observed an 83.1% accuracy during her research when classifying video based on genre. To compare results training will be done using the additional genre, with genres labeled 0-3.

**Simplified Packet Directions:** Previous research done by Campuzano and Kim [35], [36] reduced traffic trace packet directions to negative and positive "1"s depending on the direction of the data. This potentially strips the data of useful features (packet size data) that the classifier could use to determine genre and video ID. In this research training will be conducted on both: data that has not been stripped and data that has been stripped. This will give clarity on the potential usefulness of packet size data to the deep learning model.

## 4.3 Training Split

Following traditional supervised machine learning techniques, our data was split into training, testing and validation sets. Sci-Kit Learn provides a function for splitting data called "train_test_split()". Following a similar split as that used in Campuzano and Kim's research [35], [36] the data in this experiment was divided such that the training set consisted of 70% of the collected data and the testing set 30%. A validation set was used which consisted of 50% of the data in the test set. Splitting the data into subsets in this way helps reduce bias in our experiments.

Multiple training splits were made to allow for the various training scenarios listed above. The "X", or independent variables, consisted of only packet directions as this was identified by Sirinam et al. [6] as the strongest predictor for fingerprinting. The "X" dataframe was normalized to -1 or +1 in half of the experiments to test accuracy with and without packet sizes being included in the dataset. Previous research, [35], [36] use only packet directions,

and stripped the packet size from the data. However, these sizes could prove influential in classification and improve accuracy results. The "y", or dependent variable, changes based on the training scenario. When classifying videos based on Video ID, all Video IDs are stored in a separate dataframe which makes up the 1 dimensional Y dataframe. Whereas, when classifying based on genre, we use the genre column from our main dataframe to create our Y dataframe, which is made up of video genre classifications from our dataset.

After all traffic traces were read into the dataframe, the chosen training splits were made and saved into subsequent directories. The pickled training, test, and validation sets were split and saved into directories for Closed vs Open World training, and their chosen training scenario (genre with and without packet sizes; video ID with and without packet sizes).

## 4.4 Results

**Closed World:** This research looks to demonstrate how fingerprinting works against non-defended encrypted Tor traffic. The scenarios run for our closed world experiments used packet directions to classify video traces based on genre and video ID. Of these two classification features our data was also tested both with and without removing packet size from the collected directional data. With packet sizes removed, values were simplified to "-1" and "+1". Figure 4.2 shows achieved training accuracy results from our multiple experiments.

Our non-defended video genre training returned an accuracy of 86.09% with packet size included and 87.79% with packet size removed.

Non-defended training classifying video traces by video ID returned an accuracy of 54.52% with packet size included, and 60.60% with packet size removed.

To view more clearly, non-defended genre and video prediction results are as follows:

- **Genre (Packet Size Included):** 86.09% testing accuracy
- **Genre (Packet Size Removed):** 87.79% testing accuracy
- **Video ID (Packet Size Included):** 54.52% testing accuracy
- **Video ID (Packet Size Removed):** 60.60% testing accuracy

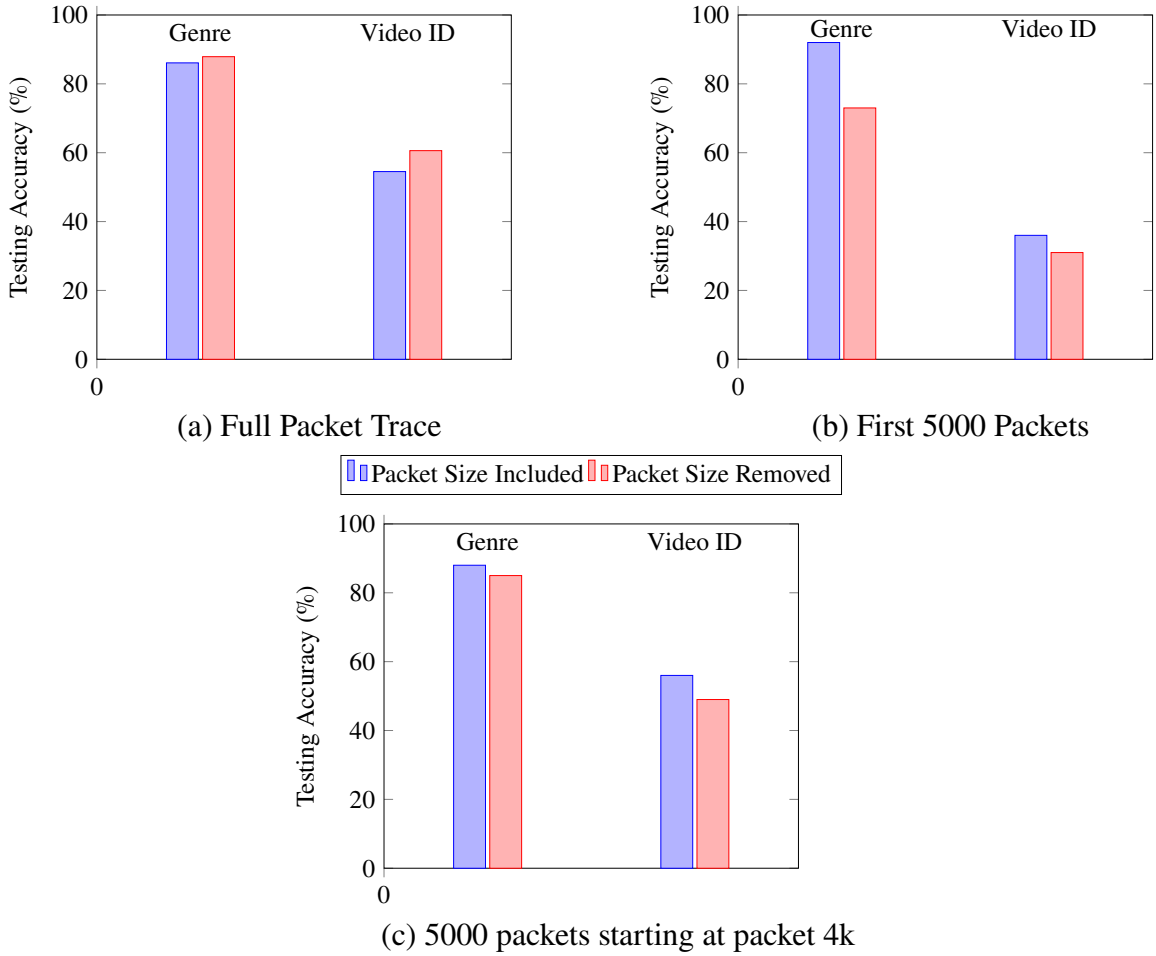


Figure 4.2. Training Accuracy Results: (a) describes full packet training on genre and video ID. (b) describes training on reduced training set including the first 5000 packets. (c) describes 5000 packet trace starting at the 4000th packet.

It was assumed that including packet size information alongside directional data would increase accuracy of the model by a significant amount. It was assumed that stripping this data would remove pertinent information that could be useful to the model for classification. However, results show a slightly higher accuracy when running models with the size information stripped from the directional data. When attempting to predict genre the difference is only that of about 1.5% which is almost negligible. However, when classifying based on video ID we see a greater difference. Accuracy with packet size included returned 54.52% compared to 60.60% with the size information stripped. This a more significant difference.

While, the dataset could still be considered small by certain measures, this still indicates a stronger correlation when size data is removed from the traffic traces.

Compared to the video ID classification results achieved in Campuzano and Kim's research, 84% and 63% respectively, we continue to observe the pattern of diminishing accuracy as our dataset grows. When including the additional genre added in this research, we observe the previously mentioned 60.60% accuracy when classifying by video ID. While accuracy continues to decrease, this illustrates a more realistic measure of video fingerprinting accuracy. Given an open world scenario, or a dataset the size of all videos that exist on the internet, classifying specific videos based on a unique id appears to a challenging task.

Sirinam et al. [6] utilized a packet sequence length of 5000 for each trace in their trials. Our research uses full packet traces to classify, however, some training was done to observe results for a smaller packet length. Attempting to use full packet traces for classification introduces potential issues. For instance how could an attacker reasonably collect a full packet trace. The attacker would need to be able to collect traffic specifically starting and ending at the beginning and ending of the video in question. Secondly, the user would have to watch the video from beginning to end with no interruption or stopping in between. This makes the use of sliced packet traces more realistic and potentially more likely to have feature rich information to influence accuracy positively.

For this reason, we simulated this scenario by using the 5000 packet length used in Sirinam et al's [6] research. This was done by taking lengthy, collected traces and slicing the data at certain packet lengths to obtain a length of 5000 for each trace. Two tests were done. The first against a set that included the first 5000 packets, and the second starting at the 4000 packet mark and keeping the following 5000 packets. This was done to see if potential ad traffic might skew the results, confuse the classifier, or provide inaccurate results.

Accuracy results for the first 5000 packets are as follows:

- **Genre (Packet Size Included):** 92% testing accuracy
- **Genre (Packet Size Removed):** 73% testing accuracy
- **Video ID (Packet Size Included):** 36% testing accuracy
- **Video ID (Packet Size Removed):** 31% testing accuracy

This model returns high accuracy results when predicting video genre, but only in the test that included packet size data. This is still a significant finding, Showing a very high accuracy score (92%) when predicting genre with a reduced packet length. We also observe a much larger difference between results with packet size removed. However, unlike the previous full packet trials, we notice the opposite effect. Preserving packet size details with a smaller packet length increases accuracy results. In the scenario where we attempt to classify videos by genre we observe an almost 20% difference between accuracy results including packet size versus when this feature is removed.

Testing accuracy results for video ID classification is influenced much more by reduced packet length. While observing an almost 30% difference in testing accuracy, it is still notable that retaining packet size information positively influences accuracy with a reduced packet length.

Testing accuracy results with a packet sequence length of 5000 and starting at the 4000 packet:

- **Genre (Packet Size Included):** 88% testing accuracy
- **Genre (Packet Size Removed):** 85% testing accuracy
- **Video ID (Packet Size Included):** 56% testing accuracy
- **Video ID (Packet Size Removed):** 49% testing accuracy

Results from this scenario more closely resemble accuracy results observed when full packet traces are included.

Plotting loss, validation loss, accuracy and validation accuracy helped provide an insight into the model as it trains over the 120 epochs used for all scenarios. Some models begin to overfit much faster than others. Figures 4.3 and 4.4 illustrate how genre and video ID training differ over the multiple training epochs. When predicting video ID, the model begins to diverge and overfit after about 30 epochs, whereas genre prediction training does not begin to overfit until much closer to the 120 epoch mark.

Comparing Figure 4.3 and 4.4, the difference between normalizing the data with -1s and +1s is evident in the amount of noise seen in the validation loss. During the latter half of training on the un-normalized genre data, there are spikes in validation loss, and a large

amount of noise. Compared to training when packet size is removed there is much less noise and the model becomes clearer to read.


Figure 4.3. Genre Prediction Accuracy and Loss


Figure 4.4. Genre (Size Removed) Accuracy and Loss

While lower accuracies are returned for video ID predictions than genre predictions, this model, represented in Figure 4.5, seems to perform well. It is evident when the model begins to overfit (around 30 epochs) which could indicate that this model will generalize better than others. Figure 4.6 shows a similar trend of less noise and relatively high accuracy when using the same data with packet sizes removed. When full packet traces are used, normalizing the data by removing and simplifying the packet size data seems to improve the models performance.


Figure 4.5. Video ID Prediction Accuracy and Loss


Figure 4.6. Video ID (Size removed) Prediction Accuracy and Loss

When reducing the number of packets included in each trace to 5000 we witness a spike in accuracy for genre prediction. This is represented in Figure 4.7. This model performs better with packet size included and returns the highest accuracy in all tests done. Video ID prediction returns much lower accuracy when using the first 5000 packets (shown in Figure 4.8), than when using the second 5000 packets (shown in Figure 4.10). There appears to be more of a correlation for genre in the first 5000 packets. Slightly less impressive results are seen when using the second 5000 packets to predict video ID and genre as shown in Figures 4.9 and 4.10 however both demonstrate that the model is able to learn given the chosen directional features. Observing our loss continuing to lower towards 0 and overfitting occurring after some number of epochs demonstrates that the model is learning and confirms our hypothesis of a strong correlation between packet direction, size and video ID and genre.


Figure 4.7. First 5000 Genre Prediction Accuracy and Loss


Figure 4.8. First 5000 Video ID Prediction Accuracy and Loss

Figures 4.11 and 4.12 show accuracy and loss for genre and video ID prediction with size information removed. These both return lower accuracy values than their counterparts with packet size data.

### Open World

Our open-world experiments are meant to simulate a scenario where an attacker has access only to a limited set of monitored videos. In the closed-world setting the attacker is assumed to have access to all possible videos viewable by the victim. This allows the attacker to train on all possible videos which is considered to be unrealistic. An attacker with a large amount of computing resources at their disposal would still be unable to obtain a traffic trace of all possible videos on the internet. This is where open-world testing and results can be helpful.


Figure 4.9. Second 5000 Genre ID Prediction Accuracy and Loss


Figure 4.10. Second 5000 Video ID Prediction Accuracy and Loss

In an open-world scenario an attacker makes use of a monitored and unmonitored dataset. The monitored dataset consists of a subset of videos an attacker would like to classify, and a set of unmonitored videos they would like to test their model against. The classifier trains on this monitored set and attempts to identify traces that fall within the monitored set from those that do not.

During Sirinam et al's [6] open-world tests ten machines were used to collect data from 5,000 sites each. This resulted in a dataset approximately 50,000 sites large. This allowed for a much more realistic open world training scenario. While we implement a similar classification process as that used in Sirinam's research, our dataset is much smaller which limits our classification abilities during the open-world experiments.


Figure 4.11. Second 5000 Genre (Size Removed) Prediction Accuracy and Loss


Figure 4.12. Second 5000 Video ID (Size Removed) Prediction Accuracy and Loss

For our experiment we used genre 3 (explosive footage) as our monitored dataset and all other genres as unmonitored/other. The data was then split into a monitored test set, unmonitored test set, a training set and a validation set. The training data consisted of genre 0, genre 1 and a subsection of our monitored data (genre 3). Our monitored and unmonitored test as well as our validation sets use the rest of our monitored data and genre 2. Using different data for our training and testing ensures the model does not return biased results from training and testing on the same data.

When evaluating results from this model we use similar methodology to that used by Sirinam et al. [6]. For each monitored video trace whose maximum output probability is greater than

some threshold, this is considered a true positive. Multiple thresholds are tested when the model is evaluated and a TPR and FPR are returned, as well as a precision and recall score.

During our testing we commonly experienced a 100% recall, showing that our model did not return any false negatives, and was able to successfully identify all instances of our monitored genre. However, the precision reached was 68% indicating a higher number of false positives. Our model successfully identified all videos in our monitored set but commonly mistook other videos for our monitored set of videos. This was consistent among all thresholds used during evaluation.

Figure 4.14 illustrates training accuracy and loss on our monitored test data with packet size included. Figure 4.15 shows the same data with packet size removed. We can see a similar trend of less noise when comparing these two scenarios as observed in the closed-world scenario. We obtain high accuracy results with both open world models. Both models return high FPRs, with lower thresholds (y-axis) returning an almost 100% FPR. However the model with packet size removed returns a precision score of 0. Our open-world testing shows that the model can successfully classify our monitored genre amongst unmonitored data, however this model could reveal more if evaluated with more data, and specifically a larger set of unmonitored video traffic traces. Evaluating this model with a much larger dataset could return more accurate and telling results. Attempting to reach a size similar to that used by Sirinam [6] would be recommended for future work in the open-world VF tests.

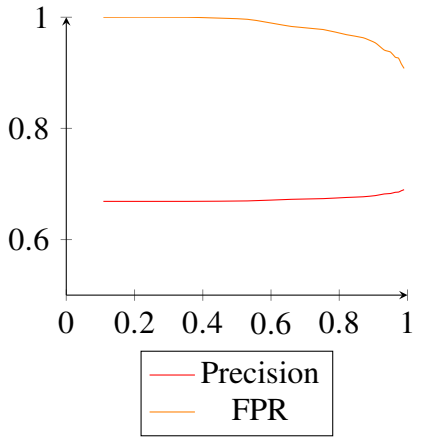


Figure 4.13. OpenWorld Threshold vs. Precision & FPR


Figure 4.14. OpenWorld Accuracy and Loss

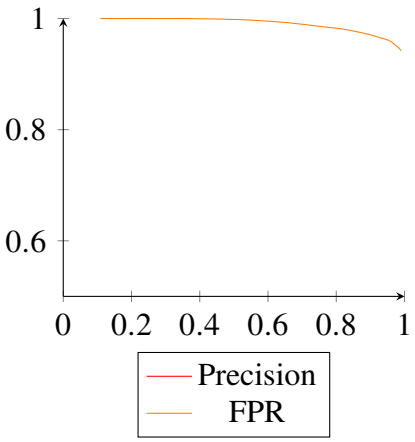


Figure 4.15. OpenWorld Threshold vs. Precision & FPR


Figure 4.16. OpenWorld Accuracy and Loss

# CHAPTER 5:
# Conclusion and Future Work

## 5.1 Future Work

In a real world scenario it is unlikely that a complete, unhindered network trace of a video viewing session will be captured. Typical users may experience ads, skip through footage or end viewing sessions before a video has completed. For this reason, it is important to test VF models in many scenarios to attempt to achieve accurate and realistic accuracy results in a range of scenarios. While results with smaller trace lengths may vary wildly from that of those with the full video trace included, it is important to see what features are influential in these situations.

Collecting data from different sites is also important in understanding the true accuracy of our model in a real world scenario. The implications of predicting video content viewed over the Tor network is most relevant in scenarios that don't necessarily include videos that are viewed by the masses. Predicting videos streamed from sites hosted by individual users, those shared on online forums and those containing content that isn't allowed on popular media sharing sites is a powerful capability if high accuracy can be observed using similar VF techniques. Comparing results from other popular video sharing sites could present an avenue of research. Creating an environment similar to that of user run Tor onion sites, and hosting videos on these sites could also prove to be a valuable source of data to train on.

Many modern websites employ techniques for combatting automated crawler traffic to their sites. This can hinder video data collection, and influence accuracy results. Many sites now present video advertisements in a convincingly random and unpredictable way. Finding ways to predict this traffic, and avoid defenses employed by modern websites will be crucial in the future to collect uninhibited video traffic.

As anonymization and encryption technologies develop, multiple defenses will be used to further obscure a user's traffic. Understanding how these defenses work and training the classifier to anticipate these traffic patterns is also an important path for future work.

Additionally, despite Tor's popularity, other anonymization tools like VPNs are widely used. Conducting VF attacks on VPN encrypted traffic could potentially apply to a larger subset of user video streaming traffic.

Other ML/DL architectures also exist that could prove successful at predicting video content with a high accuracy that could be tested on our dataset, and larger datasets. A larger and more varied dataset could also influence our open-world experiments assisting in obtaining more accurate results.

## 5.2 Conclusion

As increasingly more data passed over the internet is considered private and personal, lay users are looking for tools that provide an accessible and strong level of privacy. However, no technology is perfect. Concerns of encryption breaking and pattern matching are legitimate and must be addressed when users and organizations plan on using an online anonymity service. As seen in recent ML research and applications, ML tasks are powerful tools that can solve problems that were previously seen as unsolvable.

The aim of this study was to test the performance of VF attacks against non-defended Tor traffic in closed and open-world scenarios. Our results strongly represent the capability of these style of attacks against encrypted network traffic showing over 90% accuracy in some tests. Directional data is a strong predictor of video content and genre when given encrypted network traffic. Given the progression of encryption and network defense strategies we have identified methods for moving forward in increasing the accuracy of VF attacks against non-defended Tor traffic.

# Bibliography


[1] The Tor Project, https://www.torproject.org/.

[2] Y. D. Mane and U. P. Khot, "A systematic way to implement private tor network with trusted middle node," in *2020 International Conference for Emerging Technology (INCET)*, IEEE, 2020, pp. 1–6.

[3] G. Weimann, "Terrorist migration to the dark web," *Perspectives on Terrorism*, vol. 10, no. 3, pp. 40–44, 2016.

[4] E. Centro, E. C. Main, E. C. Links, *et al.*, "Exploring the digital jihadist underground on the Onion Router (TOR),"

[5] S. Rosenbush, "The morning download: France considers tor ban in wake of paris attacks," 2015.

[6] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, 2018, pp. 1928–1943. [Online]. Available: https://doi.org/10.1145/3243734.3243768.

[7] J. Kim and J. V. Monaco, "User Identification in Dynamic Web Traffic via Deep Temporal Features," in *2021 IEEE Security and Privacy Workshops (SPW)*, IEEE, 2021, pp. 282–290.

[8] M. Juarez, S. Afroz, G. Acar, C. Diaz, and R. Greenstadt, "A critical evaluation of website fingerprinting attacks," in *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*, 2014, pp. 263–274.

[9] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 4, pp. 436–444, 2015.

[10] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated Website Fingerprinting Through Deep Learning," in *Proceedings of the 25nd Network and Distributed System Security Symposium (NDSS 2018)*, Internet Society, 2018.

[11] E. D. Knapp, J. T. Langill, and A. G. Morris, *Cyber Arms: Security in Cyberspace*, 1st. Boca Raton, FL, USA: CRC Press, 2020.

[12] G. Strawn, "Masterminds of the arpanet," *IT Professional*, vol. 16, pp. 66–68, 3 2014.

[13] D. Vergun, *Dod work to increase cybersecurity for u.s., allies*, https://www.defense.gov/News/News-Stories/Article/Article/2351916/dod-works-to-increase-cybersecurity-for-us-allies/.

[14] T. Insights, "Number of internet of things (iot) connected devices worldwide from 2019 to 2021, with forecasts from 2022 to 2030," Transforma Insights, Tech. Rep., May 2022.

[15] E. De Cristofaro and M. Wright, *Privacy Enhancing Technologies: 13th International Symposium, PETS 2013, Bloomington, IN, USA, July 10-12, 2013, Proceedings*. Springer, 2013, vol. 7981.

[16] The Tor Project Glossary, https://support.torproject.org/glossary/consensus/.

[17] The Tor Project, *Tor directory specification*, https://gitweb.torproject.org/torspec.git/tree/dir-spec.txt, [Online; accessed 6 March 2023], Internet, 2021.

[18] A. Barton, M. Wright, and T. Shrimpton, "Towards predicting efficient and anonymous tor circuits," in *Privacy Enhancing Technologies Symposium*, Springer, 2016, pp. 76–96.

[19] P. Winter, R. Ensafi, K. Loesing, and N. Feamster, "Identifying and characterizing sybils in the tor network," in *25th USENIX Security Symposium (USENIX Security 16)*, 2016, pp. 1169–1185.

[20] J. R. Douceur, "The Sybil attack," in *International Workshop on Peer-to-Peer Systems*, Springer, 2002, pp. 251–260.

[21] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," Naval Research Lab Washington DC, Tech. Rep., 2004.

[22] M. S. Rahman, P. Sirinam, N. Mathews, K. G. Gangadhara, and M. Wright, "Tik-tok: The utility of packet timing in website fingerprinting attacks," *arXiv preprint arXiv:1902.06421*, 2019.

[23] A. Hintz, M. Almukaynizi, and S. Zeadally, "Fingerprinting websites using traffic analysis: A passive attack method on tor network," in *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, IEEE, 2016, pp. 221–224.

[24] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naı¨ve-bayes classifier," in *Proceedings of the 2009 ACM Workshop on Cloud Computing Security*, 2009, pp. 31–42.

[25] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective attacks and provable defenses for website fingerprinting," in *23rd USENIX Security Symposium (USENIX Security 14)*, 2014, pp. 143–157.

[26] M. Juárez, M. Imani, M. Perry, C. Dıaz, and M. Wright, "Wtf-pad: Toward an efficient website fingerprinting defense for tor," *CoRR, abs/1512.00524*, 2015.

[27] T. Wang and I. Goldberg, "{Walkie-talkie}: An efficient defense against passive website fingerprinting attacks," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1375–1390.

[28] M. Juarez, N. Z. Gong, A. Pal, P. Mittal, and N. Borisov, "Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, pp. 1563–1574.

[29] R. Wang, Y. Deng, Q. Xu, K. Lu, and M. Xue, "A real-time defense against website fingerprinting attacks," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2019, pp. 1625–1642.

[30] A. Panchenko, F. Lanze, A. Zinnen, M. Henze, and K. Wehrle, "Trafficsliver: Fighting website fingerprinting attacks with traffic splitting," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ACM, 2016, pp. 1388–1399.

[31] V. Shmatikov and M.-H. Wang, "Timing analysis in low-latency mix networks: Attacks and defenses," in *European Symposium on Research in Computer Security*, Springer, 2006, pp. 18–33.

[32] R. Schuster, V. Shmatikov, and E. Tromer, "Beauty and the burst: Remote identification of encrypted video streams," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1357–1374.

[33] T. Bianchi, *Most Popular Websites Worldwide as of November 2021, by Total Visits*, 2021. [Online]. Available: https://www.statista.com/statistics/1201880/most-visited-websites-worldwide/.

[34] T. Bianchi, *Most Popular Websites Worldwide in May 2022, Based on Share of Visits*, 2022. [Online]. Available: https://www.statista.com/statistics/265668/global-top-websites-ranked-by-visit-share/.

[35] C. D. Campuzano, "Towards video fingerprinting attacks over Tor," Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, 2021. [Online]. Available: https://calhoun. nps.edu/handle/10945/68304.

[36] E. S. Kim, "Predicting the unknown: Machine learning techniques for video fingerprinting attacks over Tor," Ph.D. dissertation, Naval Postgraduate School, Monterey, CA, 2021. [Online]. Available: https://calhoun.nps.edu/handle/10945/68726.

[37] J. Haferman and J. LoPiccolo, *High Performance Computing (HPC), brochure*, Brochure, 2009.

# Initial Distribution List

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California