Bayes Lower Bound Estimations Using Deep Metric Learning for Website Fingerprinting on HTTP/2 and HTTP/3

Yubo Shao

MInf Project (Part 2) Report
Master of Informatics
School of Informatics
University of Edinburgh

2025

Abstract

Website Fingerprinting (WF) attacks are causing significant concern about the users' confidentiality and privacy. Adversaries eavesdrop on victims and perform traffic analysis by passively collecting network features and using supervised learning techniques to reveal their web browsing behaviour, even if the victim is browsing in encrypted tunnels. In this paper, further investigations are done on the Bayes error lower bound estimation technique using deep learning attacks to empirically approximate the Bayes error lower bound as closely as possible, providing a mathematically proven metric for evaluating website fingerprinting techniques. Additionally, a detailed analysis of the bounds on HTTP/2 and HTTP/3 traffic is provided using different WF attacks using the Bayes error lower bound technique.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Yubo Shao*)

Acknowledgements

Massive thanks to my supervisor, Marc Juarez! Wouldn't have done it without him.

Table of Contents

Chapter 1

Introduction

1.1 Motivation

By enabling adversaries to deduce visited websites from encrypted traffic patterns, website fingerprinting (WF) attacks present serious privacy implications. Quantifying the actual difficulty of these attacks using the Bayes error rate—the theoretical minimum error any classifier can achieve—is an important objective in WF research. Practical classifiers, on the other hand, only offer empirical error rates, which can differ much from the Bayes error.

Assessing the effectiveness of defenses and the actual risk presented by WF attacks requires an understanding of and commitment to closing this gap. This project's method uses triplet fingerprinting[11] to extract embeddings that more closely resemble the real Bayes error, building on Cherubin's Bayes upper bound estimation[4].

In this project, I attempt to assess how contemporary traffic protocols and defenses affect WF attack performance by accurately approximating the Bayes error. This study sheds light on whether embedding-based techniques—such as triplet fingerprinting—offer more reliable Bayes error bound estimation assessments than conventional feature-based techniques.

In order to evaluate the security of HTTP/2 and HTTP/3 protocols as effectively as possible, the main goal of this research is to close the gap between empirically determined bounds and the real Bayes error lower bound in website fingerprinting attacks. The gap represents the difference between the theoretical minimum classification error as determined by the Bayes decision rule and actual, data-driven estimates of classifier performance. We can better understand how closely existing WF classifiers approach the theoretical limit under these protocols by reducing this disparity.

1.2 Aims and contribusions

By using deep learning techniques to improve Cherubin's estimation framework [4] and examine its applicability to modern transport protocols like QUIC and TCP, this

project's main goal is to close the gap between empirical error rates and the theoretical Bayes error lower bound in website fingerprinting (WF) attacks. The purpose of reducing this "bound gap" is to develop a better evaluation metric that is theoretically independent of classifiers and measures the intrinsic ability of encrypted traffic patterns to be distinguished under contemporary protocols such as HTTP/2 and HTTP/3.

1.2.1 Novel Contributions

This paper makes the following contributions to WF research:

- **Deep Metric Learning for Bayes Approximation**: This project is a novel application of Triplet Fingerprinting (TF)[11], a deep metric learning technique, to generate embeddings that better approximate the theoretical Bayes error lower bound[4]. This framework provides a strong, classifier-agnostic metric for assessing WF defenses, bridging the gap between theoretical security limits and empirical classifier performance.
- **Comprehensive Protocol Vulnerability Assessment**: Used Bayes error bounds to perform a methodical comparison of HTTP/2 and HTTP/3 security guarantees. Fundamental trade-offs in protocol design were uncovered by this analysis, such as TCP's vulnerability to full-trace analysis and QUIC's vulnerability to early-stage attacks.
- **Empirical Validation of Deep Learning's Theoretical Limits**: Utilizing Cherubin's bounds in deep learning adversary settings, it has shown that models based on the embeddings approach the optimal alignment with theoretical limitations ($\leq 3.5\%$ gap).

1.2.2 Key Findings

The experimental analysis yielded the following quantitative insights:

- **Deep Learning Superiority**: With 100 instances per class, Triplet Fingerprinting achieved a minimum gap of 0.38% for QUIC and 0.28% for TCP, reducing the gap between empirical error and Bayes bounds by 20–40% when compared to feature-based k-NN.
- **Protocol-Specific Risks**: At 20 percent page loading, QUIC's handshake metadata allowed for a 15 percent lower attack error rate than HTTP/2 (TF: 1.8 percent error for QUIC vs. For TCP, 2.0 percent). In contrast to TCP (0.6 percent), QUIC's full-trace error rate (1.1 percent) was elevated by its later-stage redundancy.
- **Data Efficiency**: With $\geq$50 training instances, TF stabilized embeddings ($std < 0.2\%$), demonstrating deep learning's ability to generalize from limited data—a challenge for feature-based methods like k-NN.
- **Bound Tightness**: Using ≥ 50 training instances, TF stabilized embeddings with $< 0.2\%$ standard deviation improved on k-NN's 5–20% higher variance in

low-data regimes. TF demonstrated rapid convergence with an accuracy of 85% on QUIC after just 10 instances.

- **Bound Tightness**: Bayes bounds converged to $\leq 1\%$ error for both protocols at full trace lengths, with HTTP/2 bounds being tighter (0.15% at 100% loading vs. QUIC's 0.3%).
- **Threshold-Agnostic Reliability**: The robustness of TF against threshold manipulation was validated by the near-perfect separability it maintained across protocols (ROC-AUC ≥ 0.98, PR-AUC ≥ 0.97) even with partial traces.

1.3 Project overview

The paper begins with a general introduction in Chapter 2 to the required preliminaries regarding website fingerprinting, HTTP/2 and HTTP/3 protocols, Bayes Error, my previous work, deep learning-based attacks, and a formulation to bound them following Cherubin's derivation[4].

Chapter 3 describes the experimental methodology, including the data collection process, local server hosting procedure, modification of the existing code base to accommodate HTTP/2 and HTTP/3 traffic, and evaluation methods.

Chapter 4 provides the analysis and further evaluation of the approximated Bayes error lower bound on HTTP/2 and HTTP/3 using deep learning methods in a closed-world model. Chapter 5 finishes the paper with a general summary of the work done, findings, discussions regarding the experimental limitations, potential future work, and concludes the paper.

Chapter 2

Preliminaries

2.1 Website Fingerprinting

Even though important private information like IP addresses and data payloads is somewhat shielded in encrypted network traffic, information can still be gleaned from the traffic's patterns, behaviors, and metadata. An attacker may utilize this information to train a classifier using a chosen supervised learning algorithm in the event of a Website Fingerprinting (WF) attack. This would allow the adversary to identify network traffic destinations and, in turn, disclose the victim's browsing habits. Because it only uses the traffic itself and has no connection to the actual messages or data being transferred, this type of attack gets beyond encryption systems.

Currently, there exists a variety of ML (Machine Learning) based attacking techniques focusing on the extraction of unique and insightful features of the traffic traces of a specific website to identify said website using a trained classifier, bypassing encryption. These features can be the number of incoming and outgoing packets, unique packet sizes, the time interval between sent packets, and more. Additionally, in ML-based attacking scenarios, each traffic trace is associated with a label that indicates the corresponding website, and the adversary is aware of which traffic trace corresponds to which website.

For example, LL attacks [9] employ the naive Bayes (NB) classifier for classification and count the number of packets with a specific size and direction for every feasible size and direction up to the maximum transmission unit. CUMUL attacks [10] use a Support Vector Machine (SVM) classifier for classification, and their feature set contains information on packet sequences, such as the total number of incoming and outgoing packets and the cumulative sum of packet sizes. The success of a WF attack is therefore significantly influenced by how revealing a feature set is and how well-protected the feature set is by the defender under various transport-layer protocols. Following training, the adversary can extract features from the new traces to classify a victim's network traffic, and the victim's generated traffic trace is handled as an unlabeled dataset.

To assess the performance of a WF attack, one important assumption is whether to employ an open-world or closed-world model.

With the closed-world model [4], the adversary has the greatest advantage and is

typically employed to stress the deployed WF defence or to give the adversary an idealized environment. The adversary has full knowledge of all accessible websites; hence, all websites a user may visit are monitored and identifying which monitored website the user visited is the sole task. Typically, it is assumed that the probability of a user visiting any one website in the set of all websites W is $1/n$ where $n = |W|$.

In contrast, in the open-world model [4], the adversary is only aware of a subset of all websites, hence, not all websites are monitored. The adversary's objective is to ascertain whether and which of the monitored websites are being accessed. The collection of unmonitored websites is shown in order to simulate the user accessing random unmonitored websites. The popularity of the websites is the determining factor of the probability of a website being visited. This model simulates real-world scenarios and evaluates the capability of a WF attack in authentic settings.

It is most reasonable to assume the closed-world model in the use case of this paper since maximizing the WF attacks' performance is important for obtaining the Bayes error lower bounds, as it's calculated using the empirical error rate of NN classifiers [3]. Further explanations are given in sections 2.5 and 2.8.

2.2 HTTP/2 and HTTP/3

Transport-layer protocols that are most commonly known are TCP (Transmission Control Protocol) and UDP (User Datagram Protocol). For applications that need dependable communication, TCP—the default transport protocol in HTTP/2—is recommended because of its strong error detection and correction features. Head-of-line blocking and high overhead are among its performance issues, though [13]. In HTTP/3, the IETF substituted QUIC (Quick UDP Internet Connections) for TCP as the transport protocol in order to solve these problems. QUIC replaces both TLS and TCP in key functions like encryption, multistreaming, congestion control, and dependable data delivery in its UDP-based operation [13]. Through the mitigation of TCP performance bottlenecks, this protocol typically achieves transmission efficiency comparable to or better than HTTPS [13].

Even with these improvements, QUIC still has flaws. Researchers have thoroughly examined QUIC's defenses because it replaces the TLS + TCP stack, and HTTP/2 is already vulnerable to website fingerprinting (WF) attacks. Results indicate that QUIC may be more vulnerable to WF attacks than HTTP/2 because, when QUIC is used, features that provide less information in HTTP/2 become of greater significance in improving attack accuracy. QUIC's shorter handshakes focus metadata in the first traffic [13]. Hence, QUIC is more susceptible to WF attacks because of the higher density of distinguishable features early in the transmission, which facilitates the extraction of revealing information by adversaries.

2.3 Evolution of Website Fingerprinting Features

Without access to the payload data, the need to analyze encrypted network traffic and determine which websites were visited gave rise to the field of website fingerprinting. Manually created features derived from unprocessed traffic traces were the main focus of early research in this field. The basic behavioral features of network flows were captured by pioneering studies using fundamental statistical measures like packet size distributions, inter-arrival times, and directional burst patterns [9]. These characteristics, despite their simplicity, set the stage for identifying the unique ”fingerprints” that various websites leave in network traffic. Alongside the advancement of website fingerprinting methods, feature engineering as a whole underwent substantial change. Domain-specific features that directly captured observable characteristics of network communications were initially prioritized. As machine learning methods advanced over time, researchers started utilizing more complex statistical descriptors and even automated feature extraction techniques.

A key advantage of well-engineered features is their ability to denoise raw traces and enhance data representation in the feature space. Raw network data is often high-dimensional and contaminated with irrelevant fluctuations and noise, which can obscure the underlying patterns critical for accurate website classification. By applying feature extraction techniques, engineers transform these raw traces into a reduced, more meaningful representation. This process filters out non-discriminative noise while preserving the essential characteristics of the traffic, such as temporal correlations and burst patterns. As a result, the feature space becomes more structured, enabling classifiers to operate more efficiently and with greater accuracy, even in the presence of encryption and other obfuscation measures.

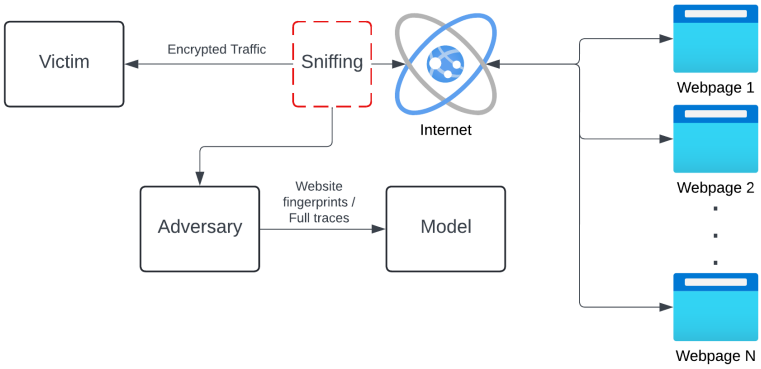
2.4 Threat model

Figure 2.1 illustrates the threat model for a website fingerprinting adversary operating within a closed-world model. During the training phase, the attacker passively captures encrypted traffic generated by a victim browsing a selection of local web pages. At the ISP or LAN level, an ML-based adversary uses the collected data to create labeled website fingerprints corresponding to each monitored page, whereas a DL-based adversary utilizes the entire traffic trace for model training.

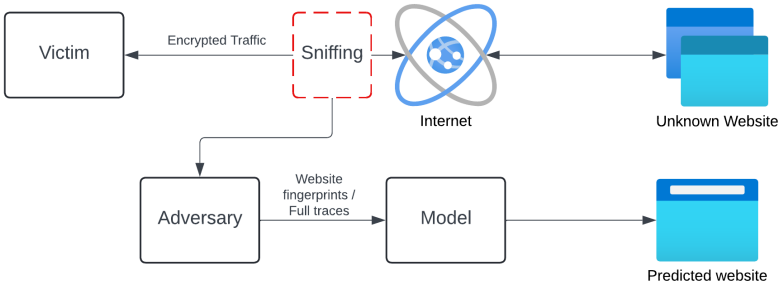
Since the fingerprinting domain is based on a closed-world model, every website visited by the victim is part of the monitored set. The attack is initiated by passively capturing fresh encrypted traffic from the victim’s local visits. For ML-based attacks, the model processes pre-extracted website fingerprints from this new traffic to predict the visited page, while DL-based attacks analyze the full traffic trace directly.

2.5 Significance of the Bayes error

The majority of previous literature evaluated the WF attack effectiveness by the accuracy against state-of-the-art defences for closed-world scenarios. Although these metrics



(a) Training



(b) Attacking

Figure 2.1: Threat model of the website fingerprinting process

offered clear quantitative insight, many contended that precision alone is insufficient for evaluating defences [8].

A strong WF defence against a poorly trained classifier would still result in low fingerprinting accuracy because the attack's accuracy depends on the classifier. For instance, if a classifier reduces the set of likely web pages that correspond to a fingerprint by mistake because of noise or confusion but is unable to accurately identify the correct page, it is not because the fingerprint does not provide enough information; rather, it is the result of a subpar classifier. This also holds true for a poorly selected feature set for an effective classifier; the low accuracy that results is not a good indicator of the feature set or the classifier's true efficacy.

A different way to evaluate the performance of WF attacks is using the Bayes error which provides a mathematical error lower bound for a WF attack technique [4], where the overlapping region of the distributions between each of the gathered attributes and their frequencies across the intercepted data for every web page is used to calculate the ”smallest error achievable.” This is a stronger indicator of the maximum potential of an attacking technique and is theoretically classifier-independent. One important note is that **theoretically**, a true Bayes classifier, and hence its Bayes error, should be feature-independent too (a classifier trained on any different form of information

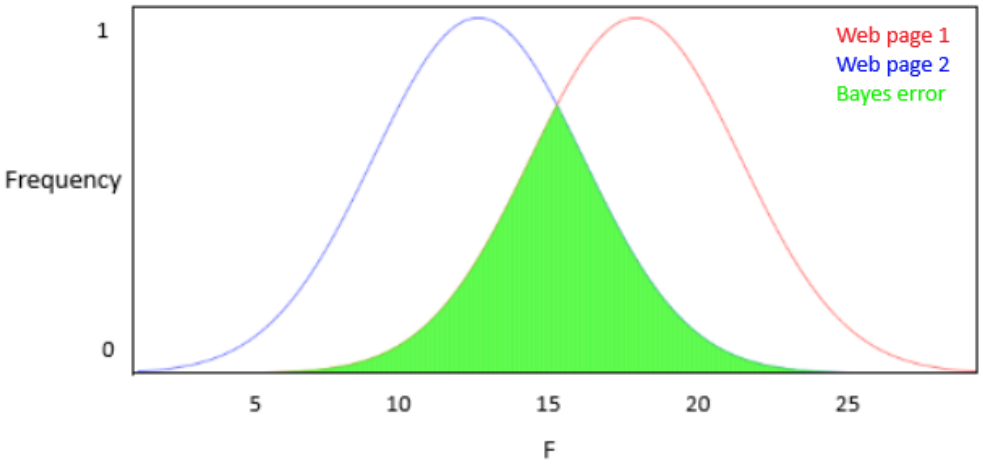


Figure 2.2: An intuitive example distribution for feature F and its frequency

obtained from the original data shouldn't have improved accuracy over the original) [4]. This does not hold true empirically, and a more revealing feature set will have a significant impact on accuracy, however, the lower bounds of the true Bayes error can be approximated mathematically. This is further explained in section 2.8.

Figure 2.2 is a rudimentary example of Bayes error modified from my previous year's work:

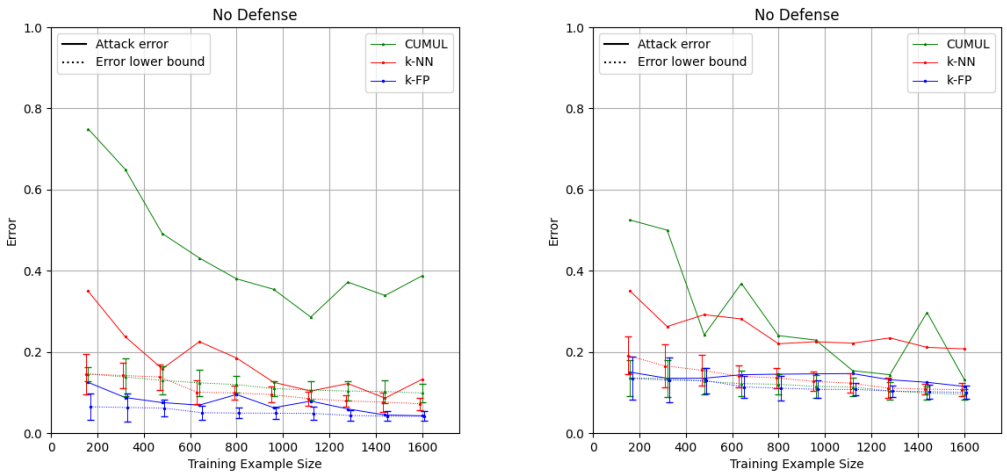
- Assume the adversary can observe only a single feature, F.
- Consider a scenario with only two web pages.
- The relationship between feature F and its frequency for each web page can be modeled as a probability distribution.
- The Bayes classifier predicts the destination web page based on which page has the higher frequency at a given value of F.
- For instance, if $F = 20$, the Bayes classifier would predict web page 1 if it has a higher frequency at that value.
- The Bayes error represents the minimum probability of misclassification, corresponding to the area where the classifier's prediction is incorrect.
- In practice, this process generalizes to any number of web pages and features, with the Bayes error calculated from the observed data.

With this knowledge, it is evident that the empirical attack error would always be above the theoretical Bayes error, which itself would always be above the estimated Bayes error based on empirical data. This provides a minimal performance guarantee or an expected performance metric for any WF attack, which is a much better indicator than accuracy itself.

2.6 Previous work

In my previous year's work, I investigated the WF attack performance against undefended and fixed-rate defended traffic traces of HTTP/2 and HTTP/3. I used the WCN+ dataset to replicate Cherubin's results and validate his conclusions [4], after which I experimented with my personally collected dataset in terms of attack failure rates (error rates) and Bayes lower bounds.

While I found that attacks generally performed marginally to significantly better on HTTP/3 in various scenarios, it also intrigued me as I saw the distances between the empirical error rate and estimated Bayes error for various WF attacks are quite far. From my previous year's experiment results as shown in Figure 2.3, while k-FP [6] performed extremely close to the estimated Bayes error lower bound, often within the standard deviation of the bound, it is apparent that for k-NN [3] and CUMUL [10] there exists much performance to be desired. The distance between the empirical error rate and the estimated bound can be as high as five to ten times the standard deviation of the estimated bound.



(a) HTTP/2 (b) HTTP/3

Figure 2.3: Lower bound $\hat{R}^*$ and attack error rate $\hat{R}^A$ on collected HTTP/2 and HTTP/3 traces (Closed-world) with respect to varying sizes of training examples Z_{train}. (Lower bound markers are slightly offset to better present the standard deviations.)

One thought is that we can attribute this phenomenon to the limitations of experiments I mentioned at the conclusion of last year's work. However, as an extension to last year's experiment, I attempt to better approximate the Bayes error using deep learning methods and reduce the distance between the real and the estimated Bayes error as much as possible. It would provide a more accurate evaluation for all WF attacks, regardless of the classifier, whether it's ML-based or DL-based (deep learning based), or the feature set (if distances between approximated lower bound and empirical error rates of attacks are sufficiently minimal).

A critical observation from these experiments is that lower-bound estimations derived

from different attacks (e.g., k-FP, CUMUL) do not fully overlap. For instance, while the CUMUL attack's empirical error rate is bounded by its estimated Bayes error, CUMUL's range does not intersect with the range between the bound and the empirical error rate of k-FP (Figure 2.3). This raises a theoretical concern: if all bounds approximate the same true Bayes error, why don't their ranges overlap? The resolution lies in Cherubin's assumption that bounds converge to the true Bayes error only as the number of training instances per class $n \to \infty$[4]. In practice, finite datasets introduce variance in feature representation and posterior estimation, leading to non-overlapping bounds. However, empirical results (Figure 2.3) demonstrate that as training samples increase, bounds progressively tighten and converge toward a common limit. This suggests that with sufficient data, discrepancies between bounds diminish, validating their role as approximations of a singular theoretical limit. Thus, non-overlapping bounds in small-sample regimes reflect estimation noise rather than multiple Bayes errors, reinforcing the need for large-scale datasets to stabilize bounds.

With the above considerations, my previous work has several shortfalls:

- The lack of different types of WF attacks, especially deep-learning-based WF attacks, doesn't rely upon feature-extracting algorithms;
- A relatively small data set that causes the estimations to be less precise and lower bounds estimations to be looser;
- While the Bayes error lower bound estimations bound the actual theoretical Bayes error lower bound, the distance between them is unknown, and the range is fairly wide;
- The lack of hyperparameter tuning, the conducted experiment only used the default values.

Hence, the objectives of this paper are to investigate the extent how close the real bounds can be approximated with:

- A deep-learning-based state-of-the-art WF attack such as Triplet Fingerprinting[11], using its existing tuned hyperparameters, which is introduced in section 2.7;
- A larger dataset with 10000 unique traffic traces over 100 websites, five times last year's 2000;
- Variance considerations of Triplet Fingerprinting to further reduce uncertainty.

2.7 Previous deep learning attacks & Triplet Fingerprinting

Since DL-based attack techniques are used, this section details the differences and unique traits employed by various well-known DL-based attacks and specifically Triplet Fingerprinting which is used for the purpose of this dissertation. Firstly, The key differences between ML and DL-based attacks are as follows:

- **Feature Extraction:** ML-based attacks require manual feature extraction, while DL-based attacks automatically learn features from raw traffic data.
- **Performance and Accuracy:** DL-based attacks tend to outperform ML-based attacks, especially in complex, noisy environments, due to their ability to capture deeper trends in the data.
- **Computational Resources:** DL-based attacks, especially those using CNNs or more advanced architectures like Var-CNN, require significantly more computational power for training and inference compared to ML-based models.
- **Scalability:** DL-based approaches can scale better with larger datasets and more websites, whereas ML models like k-NN and k-FP may struggle with larger training sets or more complex feature spaces.

Different from the examples and attacking procedures of ML-based attacks I've introduced in section 2.1, DL-based attacks operate slightly differently:

2.7.1 Deep Fingerprinting (DF)

DF [12] is a pioneering deep learning model introduced to enhance website fingerprinting attacks by leveraging the capabilities of Convolutional Neural Networks (CNNs). The main advantage of DF is that it can operate on raw or minimally preprocessed traffic data, automatically learning the most discriminative features, which eliminates the need for manual feature engineering.

CNNs, the main component of DF's architecture, are effective at identifying patterns and spatial hierarchies in data. Although CNNs are frequently employed in computer vision, their architecture can also be applied to the analysis of sequential data, such as traffic traces. CNNs in DF automatically record global patterns, like the general behavior of a session, as well as local patterns, like brief packet bursts. DF performs better than conventional models at identifying websites because of its capacity to capture multi-scale features.

DF is quite effective in real-world applications since it has demonstrated strong robustness in the presence of noise and variability, such as various browsing behaviors, varied connection conditions, and multiple open tabs.

2.7.2 Variational Convolutional Neural Network (Var-CNN)

Building on the advantages of DF, Var-CNN [1] (Variational CNN) uses variational techniques to enhance the model's generalization across a greater range of traffic conditions. This technique addresses a major flaw in traditional CNN models by better capturing the variance and randomness in website traffic. Due to varying browsing habits, network conditions, and browser-specific behaviors, network traffic naturally varies. Despite their strength, CNNs in DF may find it difficult to generalize effectively when there is this kind of variability; This is where Var-CNN shines.

The concepts of variational inference, which are frequently employed in variational autoencoders, are incorporated into Var-CNN. This method aids the model in capturing

the variability of traffic traces in addition to the mean patterns. In essence, the model learns to model the distribution of features across various traffic traces in addition to learning fixed features for each class (website). This increases the model's resilience in situations with noisy or highly fluctuating traffic.

By learning to capture not just fixed patterns but also the variability in traffic, Var-CNN can generalize better to unseen traffic traces. This makes it more effective in real-world environments where noise and randomness are common. Additionally, the variational aspect acts as a form of regularization, preventing the model from overfitting to specific training data. This allows it to maintain high accuracy even in new conditions or with websites not seen during training.

2.7.3 Triplet Fingerprinting (TF)

Triplet Fingerprinting (TF)[11] is a specialized website fingerprinting approach that leverages deep metric learning to produce embeddings from raw traffic traces. This method extends the capabilities of prior deep learning-based models by focusing on learning a structured embedding space, where the relative distances between traffic patterns capture their similarities and differences. Unlike direct classification models like Deep Fingerprinting (DF) or Variational Convolutional Neural Networks (Var-CNN), TF produces an embedding that can be used with traditional classifiers like k-Nearest Neighbors (k-NN) for more flexible and interpretable downstream tasks.

2.7.3.1 Architectural Integration with DF

The convolutional structure of DF serves as the foundation for TF, which reuses its feature extraction layers while redefining the output layer to generate embeddings rather than classifications. Through its convolutional layers, TF is able to inherit DF's strengths in capturing both local and global traffic patterns thanks to this architectural choice. Without the need for manual feature engineering, the extracted embeddings function as compressed representations of raw traffic traces, keeping crucial details about the observed traffic.

The objective function is where the main distinction is found. TF uses a triplet loss function, whereas DF uses the cross-entropy loss to directly classify inputs. In the embedding space, this loss pushes the model to map dissimilar traffic traces farther apart and similar ones closer together. TF improves generalizability to new traffic patterns and unseen websites by optimizing for relative distances rather than absolute labels.

2.7.3.2 Triplet Network and Metric Learning in TF

TF adopts a triplet network structure [7], which consists of three parallel CNNs sharing weights to process three input samples simultaneously:

- **Anchor (A)**: The reference traffic trace.
- **Positive (P)**: A trace from the same class (website) as the anchor.
- **Negative (N)**: A trace from a different class (website) than the anchor.

The model is trained to minimize the distance between the anchor and the positive while maximizing the distance between the anchor and the negative. Formally, the triplet loss L is defined as:

$$L = \max(d(f(A), f(P)) - d(f(A), f(N)) + m, 0)[7]$$

where $f(\cdot)$ represents the model's embedding function, d is the Euclidean distance, and m is a margin that enforces separation between positive and negative pairs.

The triplet loss method of metric learning makes it possible for the model to identify intricate patterns in the traffic traces while guaranteeing that comparable samples group together. This produces an embedding space where semantic similarity between traffic flows is directly reflected by proximity, which makes the embeddings appropriate for k-NN classification.

2.7.3.3 Embedding Extraction and k-NN Classifier Usage

The convergence of bounds under large datasets informs the design of deep learning attacks like TF. Unlike feature-based methods (e.g., k-NN), which rely on handcrafted features vulnerable to finite-sample biases, TF's embedding space learns protocol-agnostic representations that better approximate the true data distribution. In my experimental setup, raw traffic traces are first processed by the TF model to generate embeddings. These embeddings are then used to train and evaluate a k-NN classifier. This setup allows for both:

- **General k-NN Classification:** Evaluating the performance of embeddings across arbitrary k values, providing insight into the stability and accuracy of the learned feature space.
- **1-NN for Bayes Error Estimation:** Using 1-NN classifier results to approximate the Bayes error lower bound using equation 2.4 in section 2.8.2, as it represents the theoretical minimum classification error for the given distribution of data.

For both local and global traffic structures, the consistency of the embedding space is evaluated by comparing the results of lower bounds and k values. This method assesses the degree to which the embedding-based model accurately captures the true Bayes error in actual traffic situations. More information is explained in Chapter 3.

2.7.3.4 Alignment and Uniformity in Embedding Spaces

A critical aspect of TF's performance lies in the balance between *alignment* and *uniformity* in the embedding space:

- **Alignment:** Ensures that embeddings of traffic traces from the same class are closely grouped together. Effective alignment improves classification accuracy by reducing intra-class variance.
- **Uniformity:** Ensures that embeddings are evenly spread across the feature space, minimizing overlaps between different classes. Excessive uniformity can reduce

the model's discriminative ability by collapsing embeddings of dissimilar classes into a shared region.

By bringing together similar samples and pushing apart dissimilar ones, TF's triplet loss optimizes for both goals. Overfitting to the training distribution may result from excessive alignment, while an overemphasis on uniformity may result in the loss of class-specific information. Sustaining this equilibrium is essential for precise categorization of both known and unknown traffic patterns.

2.8 Bounds on deep learning-based adversaries

In this section, I provide an alternate formulation of Cherubin's proof [4], which I find significantly easier to understand, on the inequality between estimated Bayes error lower bound, theoretical Bayes error, and empirical error. The original work was only formulated for ML-based attacks and bounded them; however, by the definition in Cherubin's work, the inequality should apply to DL-based attacks as well. Hence, the following formulations are written to include both ML-based and DL-based adversaries.

2.8.1 Cherubin's Bound and Bayes Optimal Classifier

Based on the definition given by Cherubin [4], I used the majority of the definitions for ML attacks on all attacks, as the proof provided by Cherubin wasn't derived from any particular classifier or specific mechanics. This means that their performance should still be bounded by the same Bayes limit that applies to k-NN or k-FP attacks, and my formulations are sensible mathematically.

The definitions used in this derivation, which is based on materials from "Pattern Recognition and Machine Learning" by Christopher Bishop [2], are shown below.

Cherubin's derivation revolves around the probability of misclassification [2]:

$$P_e = 1 - \int_{\mathcal{X}} \max_{y \in \mathcal{Y}} P(y|x)\, dx \tag{2.1}$$

where:

- P_e is the Bayes error rate or the minimum error rate achievable by any classifier.
- $P(y|x)$ is the **posterior probability** of class y given the feature vector x (i.e., the feature vector).
- $\mathcal{X}$ is the feature space (all possible traffic traces).
- $\mathcal{Y}$ is the set of possible classes (the different websites).

This formula should represent the theoretical lower bound for any attack, including machine learning-based or deep learning-based.

2.8.2 Application to Deep Learning Attacks

A simpler understanding of deep learning models like DF or Var-CNN can be viewed as more powerful classifiers that attempt to approximate the true posterior distribution $P(y|x)$ more effectively than classical ML models (like k-NN or k-FP). However, they still rely on the same underlying data (traffic traces), and thus their performance is limited by the same theoretical lower bound.

Cherubin's bound can apply to deep learning-based models as such:

Feature Space $\mathcal{X}$. For both classical ML and deep learning models, $\mathcal{X}$ represents the traffic features such as packet sizes, timings, and directions. In deep learning models like DF or Var-CNN, the feature space is often expanded to include more granular or abstract representations learned by the network layers. Theoretically, the feature space doesn't change between models, but deep learning methods can explore it more effectively.

Posterior Probability $P(y|x)$. The deep learning models aim to approximate $P(y|x)$ with higher accuracy by learning complex patterns in the data. While deep learning models may reduce empirical error rates compared to classical models, they are still subject to the uncertainty in the posterior distributions, which governs the theoretical error bound.

Hence, for the theoretical derivation of DL-based attacks, we can think of them as function approximators of $P(y|x)$ more accurately than ML-based methods. We can define the posterior distribution learned by a deep network as:

$$P^{DL}(y|x) \approx f_\theta(x) \tag{2.2}$$

where $f_\theta(x)$ is the output of the deep learning model (parameterized by the set of parameters θ)) that approximates the probability distribution over classes given the feature vector x. Rather than substituting the DL posterior probability to equation 1, the empirical error which is our interest can be simply expressed as:

$$\hat{P}_e^{DL} = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\{y_i \neq \hat{y}_i\}} \tag{2.3}$$

Where:

- $\hat{y}_i$ is the predicted class for the i-th sample.
- y_i is the true class label.
- n is the number of samples.
- $\mathbb{1}$ is an indicator function that counts the misclassified samples.

And since the theoretical minimum P_e still applies:

$$\hat{P}_e^{DL} \geq P_e \geq \hat{P}_e \tag{2.4}$$

where $\hat{P}_e$ is the empirical lower bound calculated as such given by Cherubin [4]:

$$\hat{P}_e = \frac{L-1}{L}\left(1 - \sqrt{1 - \frac{L}{L-1}\hat{P}_e^{NN}}\right) \tag{2.5}$$

Where:

- L is the number of all possible classes $|\mathcal{Y}|$ (# of the different websites).
- $\hat{P}_e^{NN}$ is the empirical error of the NN classifier computed on the same dataset.

I should stress that **the inequality that is Equation 2.4 only applies when** $n \to \infty$.

2.9 Related work

A Bayesian framework for assessing the security of website fingerprinting (WF) defenses was introduced by Cherubin [4]. The authors derive theoretical bounds on the effectiveness of WF defenses by modeling realistic adversarial settings, in contrast to traditional methods that frequently assume perfect classifiers. They contend that against powerful adversaries who use Bayesian inference instead of straightforward, naïve classifiers, many defenses might only offer a limited level of protection. The main contribution is the introduction of a more realistic threat model, which aids in determining the actual strength of WF defenses under various knowledge-based assumptions made by the attacker.

Sirinam et al. [12] presented Deep Fingerprinting (DF), a powerful WF attack using Convolutional Neural Networks (CNNs). This approach surpasses conventional machine learning models by autonomously learning traits from unprocessed traffic data, eliminating the need for manual feature engineering. The research reveals DF's exceptional efficiency in recognizing websites, even when defensive measures such as traffic obfuscation or padding are implemented. The authors substantiate that DF can significantly surpass prior methods in both closed-world and open-world settings, underscoring the potential of deep learning techniques to overcome numerous existing WF defences and substantially increase the effectiveness of attacks.

Sirinam et al. [11] introduced Triplet Fingerprinting (TF), a novel website fingerprinting attack leveraging triplet networks and N-shot learning to enhance practicality and portability. TF achieves high accuracy with few samples, in contrast to earlier approaches that needed large amounts of training data that were updated frequently. Interestingly, it remains effective even when network conditions and time intervals between training and testing data change, attaining 85% or greater accuracy even when data is gathered on different networks years apart. Users of anonymity systems like Tor may have serious privacy concerns about this method because it shows that accurate website identification is possible with little data and processing power.

Deng, Li, and Xu [5] introduced Holmes, an advanced website fingerprinting (WF) attack that leverages spatial-temporal distribution analysis to identify websites during the early stages of page loading. The robustness and dependability of WF attacks under

changing network conditions and different defenses are increased by Holmes' efficient correlation of early-stage traffic with complete traffic profiles using adaptive data augmentation and supervised contrastive learning. In comparison to nine deep learning-based WF attacks currently in use, Holmes improves the F1-score for early-stage traffic identification by an average of 169.18%, according to evaluations conducted across six datasets. The effectiveness of early-stage WF attacks has significantly improved since Holmes was able to identify websites in real-world scenarios involving dark web traffic when only about 21–71 percent of the page had loaded.

Chapter 3

Experimental Methodology

3.1 Data Collection & Preprocessing

The experimental dataset was constructed to satisfy the following four requirements for evaluating QUIC vs HTTPS vulnerabilities under WF attacks:

- **Scale:** 100 traces per website provide sufficient feature density for machine learning analysis
- **Realism:** Encrypted traffic patterns mirror actual user browsing behavior
- **Containment:** All resources originate from intentional navigation paths
- **Adversarial Perspective:** Collection methodology replicates passive network eavesdropping

The testbed architecture adapts Zhan et al.'s controlled network approach [13] with key modifications for protocol isolation. As shown in Figure 3.1, traffic generation occurs through automated browsing of Alexa's top 100 English-language websites[1] that support both HTTP/2 and HTTP/3. This selection criterion ensures there is a direct comparison between 10,000 TCP (HTTP/2) and 10,000 QUIC (HTTP/3) traces, and additionally, the Alexa rankings reflect actual user visitation patterns more accurately than the academic pages I used in the previous year's work.

A headless Chrome instance generated traffic through sequential page loads, with separate profiles for HTTP/2 and HTTP/3 configurations. Each website was loaded 100 times per protocol using Selenium automation, with randomized inter-request intervals between 2-5 seconds to simulate human browsing patterns.

This methodology improves upon prior university-focused datasets from last year by increasing scale 5x (100 vs 20 sites) while maintaining temporal consistency - all traces were collected within a 14-day period using fixed browser/OS versions to minimize versioning artifacts. The final raw data comprises 20,000 fully labelled traces (100 sites × 100 traces × 2 protocols) stored as encrypted `.pcap` files with associated metadata.

[1] https://www.expireddomains.net/alexa-top-websites/

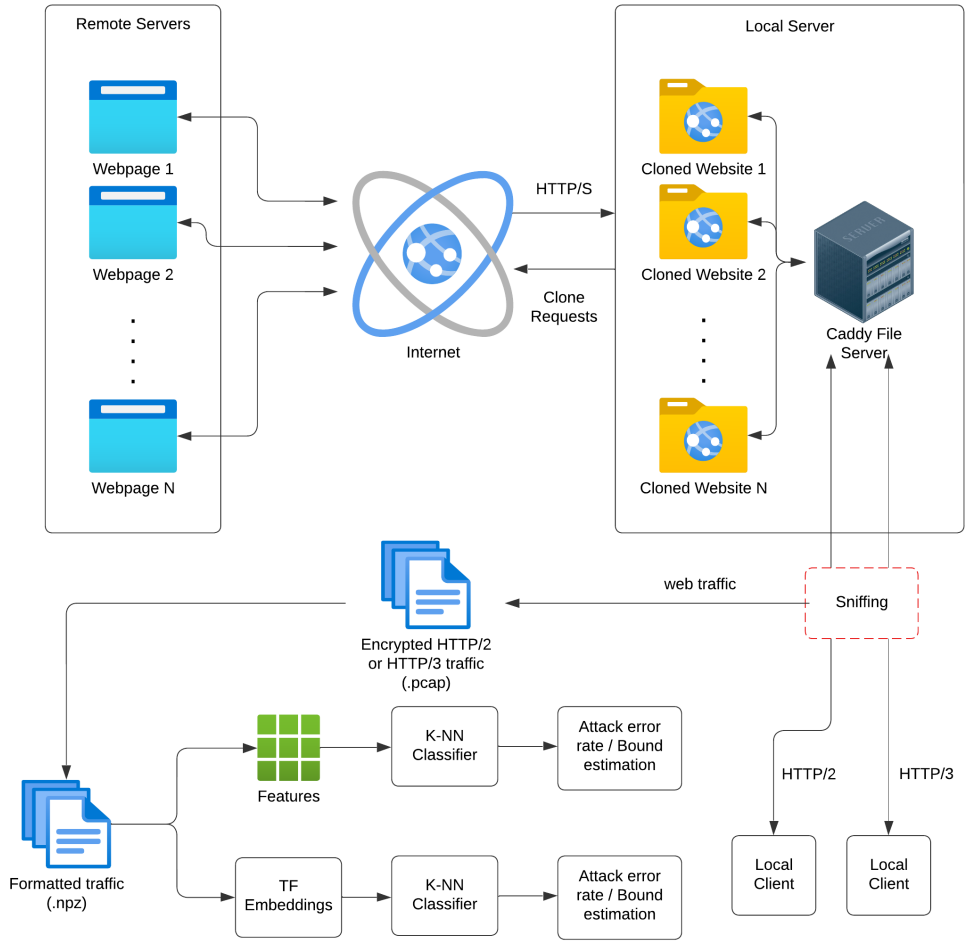


Figure 3.1: Overall Testbed Architecture

The following automated and manual validation procedures were implemented to ensure trace quality:

- `tshark` analysis for protocol headers (QUIC long headers for HTTP/3, TCP/TLS for HTTP/2)
- Encryption confirmation (TLS 1.3 handshakes, QUIC Initial packets)
- Session termination verification (`FIN/RST` flags in TCP, QUIC `CONNECTION_CLOSE`)
- Monotonic timestamp enforcement ($\forall t_i < t_{i+1}$)
- Empty packets in traces and failed-to-load traces (session terminated successfully but page load failed, determined using trace length).
- Retransmission detection (TCP dup ACKs > 3, QUIC `RETIRE_CONNECTION_ID`)

Lastly, the data are preprocessed to accommodate the code bases which are further detailed in section 3.2:

- The data are formatted to an NPZ file with the following structure:

  - `X`: 2D array of shape $(N, 5000)$ containing normalized sequences
  - `y`: 1D array of integer class labels
- Each sequence was truncated or zero-padded to a fixed length of 5,000 packets to standardize input dimensions.
- The data was then shuffled to remove inherent ordering biases while retaining label consistency.

3.2 Experiment frameworks

3.2.1 Website-Fingerprinting-Library

The experimental implementation leverages the `Website-Fingerprinting-Library`, an open-source PyTorch framework originally designed for Tor traffic analysis[5]. The repository provides modular implementations of state-of-the-art deep learning models for website fingerprinting, including Triplet Fingerprinting (TF). Additionally, despite the framework's Tor-centric design, it accepts temporal-directional data as inputs, making it directly compatible with QUIC/TCP traffic.

The `Website-Fingerprinting-Library` can be easily adapted for this project's purpose because of the following existing implementations:

- **Triplet Network Architecture**: The framework mirrors exactly the original Triplet Fingerprinting [11] implementation with:
  - Shared-weight CNNs processing 5,000-element directional sequences, which is also compatible with directional timestamp sequences;
  - And Triplet loss with margin $m = 0.2$
- **Parameter Flexibility**: The framework implementation also allows accessible hyperparameter change in their pipeline:
  - Adjustable k-values in `model_utils.py` for the final k-NN classification of TF
  - ROC-AUC/PR-AUC metrics can be easily added to `evaluator.py`
  - The training/testing set size can be easily set in `data_split.py`

3.2.2 Website-Fingerprinting-Evaluation-Suite (WFES)

For feature-based analysis similar to my previous year's work, the same modified version of WFES[4] was employed. Key adaptations from the original Tor-focused implementation to accommodate QUIC/TCP traffic include:

- Extended feature vectors with $\pm$size encoding (vs. Tor's fixed 512B cells)
- Cumulative size histograms per transmission direction (given non-fixed packet sizes)

The WFES is modified to accept NPZ files as inputs with identical train/test splits as TF through predefined indices on command line. The modified WFES implementation also maintains backward compatibility with standard k-NN benchmarks with Tor while being able to process HTTP/2 and HTTP/3 sequences. All experiments use scikit-learn's `NearestNeighbors` implementation with the Euclidean distance metric, $k = 1$ for Bayes error estimation, and $k = 5$ for the default classification.

3.3 Evaluation Metrics & Scenarios

3.3.1 Experimental Scenarios

To systematically assess the raw accuracy and robustness of website fingerprinting (WF) attacks under realistic adversarial constraints, two experimental scenarios were designed to evaluate classifier performance under partial traffic observation and limited training data. These scenarios provide insight of the unique characteristics of each protocol and their behavior against adversaries in real-world settings, where network monitoring may be intermittent or resource-constrained.

3.3.1.1 Partial Traffic Observation (Page Loading Ratio)

Objective: This scenario evaluates how early in a browsing session sufficient discriminative features emerge to enable accurate website identification for both QUIC and TCP. It simulates adversaries who intercept traffic during the initial phases of page loading (e.g., due to connection drops, time-limited surveillance, or computational constraints).

Implementation Details:

- Traffic sequences were truncated at incremental observation points corresponding to 20%, 30%, 40%, 50%, and 100% of total packet counts.
- Truncated traces were padded with zeros to maintain a fixed input dimension of 5,000 packets, ensuring compatibility with deep learning architectures.
- A stratified 90%/10% train/test split preserved class distributions. To mitigate temporal biases, traces were shuffled prior to splitting, ensuring no temporal correlation between training and testing data.

Motivation: Critical metadata is concentrated by modern protocols such as QUIC (e.g., in early packets (connection IDs, encryption parameters), which even with only partial observation may reveal recognizable patterns. Secondly, defenses like dynamic content loading and traffic padding frequently give priority to later-stage traffic, which makes early packets less obscured. Finally, adversaries may give partial traces priority in order to minimize computational overhead in real-time surveillance systems in terms of operational efficiency.

3.3.1.2 Data Efficiency (N-Shot Learning)

Objective: This scenario evaluates classifier generalizability under limited training data, reflecting how generalizable the traces of each protocol are and their performance

against adversaries who cannot collect large labeled datasets (e.g., infrequently visited websites or ephemeral services).

Implementation Details:

- Each class were given $N \in \{10, 20, 50, 100\}$ instances per website, the test set remained fixed at 20% of the given traces.
- Subsampling preserved the proportional representation of all 100 websites, and the traces are randomly selected to mitigate sampling bias.

Motivation: Tests whether handcrafted features (k-NN) or automated embeddings (TF) better generalize from sparse data. Secondly, it validates whether state-of-the-art ML-based and DL-based attacks remain viable against low-traffic websites or short-term monitoring campaigns.

3.3.2 Performance Metrics

Three complementary metrics were selected to holistically evaluate attack efficacy, theoretical bounds, and threshold-agnostic robustness:

- **Attack Error Rate**: Calculated as the percentage of misclassified test traces (Error rate = 1 - Accuracy). It directly quantifies attack failure rates under specific experimental conditions. Lower values indicate stronger attacks. However, using accuracy alone may mask class-specific vulnerabilities or threshold-dependent performance, necessitating supplementary metrics.
- **Bayes Error Bound Estimation**: Calculated using the 1-Nearest Neighbor (1-NN) classifier on learned embeddings, following Cherubin's inequality (See equation 2.4 in section 2.8.2). The the lower bound estimation made with 1-NN error rate provides an asymptotically unbiased estimate of the Bayes error as $n \to \infty$. For finite datasets, it serves as a practical lower bound for classifier-agnostic evaluation.
- **ROC-AUC (Receiver Operating Characteristic)**: Calculated by plotting the True Positive Rate (TPR) against the False Positive Rate (FPR) across all classification thresholds and computing the area under this curve, this is done directly using the `scikit-learn` package. It measures separability across all classification thresholds, with 1.0 indicating perfect discrimination.
- **Precision-Recall AUC (PR-AUC)**: Derived by plotting Precision against Recall across thresholds and measuring the area under this curve, this is also done directly using the `scikit-learn` package. It evaluates precision-recall trade-offs under class imbalance, where high recall at low false-positive rates is essential. PR-AUC is more informative than ROC-AUC when negative classes dominate (e.g., open-world attacks with many unmonitored sites), but in this project, the purpose of AUC values is to validate that empirical error rates are not artifacts of threshold cherry-picking (with AUC ≥ 0.95), ensuring reliable comparisons between protocols and classifiers.

Chapter 4

Evaluation

4.1 Variance

This section evaluates the stability and generalizability of Triplet Fingerprinting (TF) embeddings under varying experimental conditions. By analyzing variance across training regimes and partial traffic scenarios, we assess the robustness of deep metric learning in approximating the Bayes error lower bound.

4.1.1 Dataset Preparation

The evaluation employed a custom dataset comprising 20,000 encrypted traces (100 websites × 100 traces × 2 protocols). To ensure reproducibility and isolate protocol-specific effects:

- **Stratified Splitting**: A fixed 70/30 train/test split was applied. The 70/30 split is to ensure a sufficiently large testing set to decrease measurement noise. In actual experiments, the previously mentioned splits are used instead.
- **Deterministic Trials**: Three independent trials per protocol were conducted with fixed random seeds to control for initialization and ordering biases. GPU acceleration was disabled to eliminate hardware-induced variance.

To enforce determinism, All `random` seeds were fixed, and GPU acceleration was disabled. Dropout remained active during training but was not used during inference as the inference is done by the k-NN classifier. This design minimizes extraneous noise, ensuring observed variance stems from protocol characteristics or model dynamics rather than experimental artifacts.

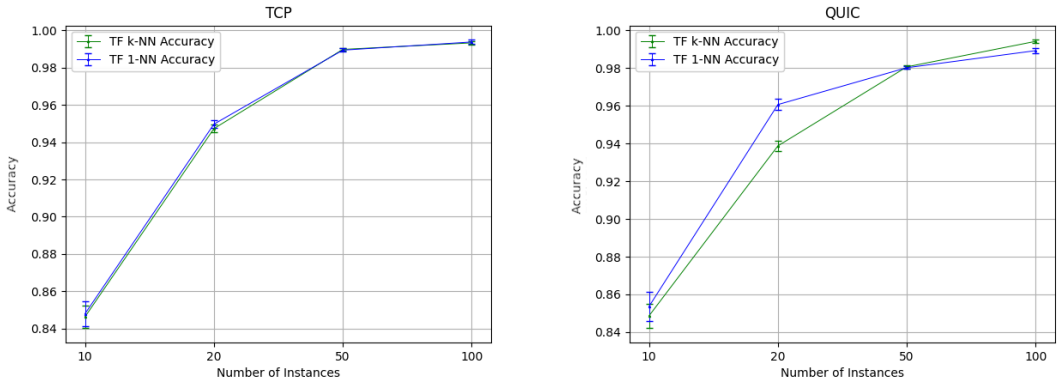
4.1.2 N-Shot Learning (NS) Experiments

The N-Shot experiments (Figure 4.1) quantified how training instance scarcity impacts embedding stability. observations include:

Low Data Regimes (10–20 Instances):

- **QUIC**: At 10 instances, TF achieved 85% accuracy with a standard deviation (std) of 0.5–1% (Table 4.7). The higher variance reflects sparse intra-class clusters in QUIC's embedding space, where early metadata concentration creates overlapping distributions for under-sampled classes.
- **TCP**: With 10 instances, TCP exhibited slightly higher std (0.6–1.2%) due to its sequential dependency structure. Longer packet sequences require more data to capture temporal patterns, amplifying variance in small-sample regimes.

High Data Regimes (50–100 Instances): Both protocols achieved sub-0.2% std at 50+ instances, demonstrating TF's data efficiency. QUIC's std decreased faster (0.5% $\rightarrow$ 0.1% from 20$\rightarrow$50 instances) than TCP's (0.7% $\rightarrow$ 0.2%), as its feature-dense handshake packets enabled rapid cluster stabilization.



(a) HTTP/2 (b) HTTP/3

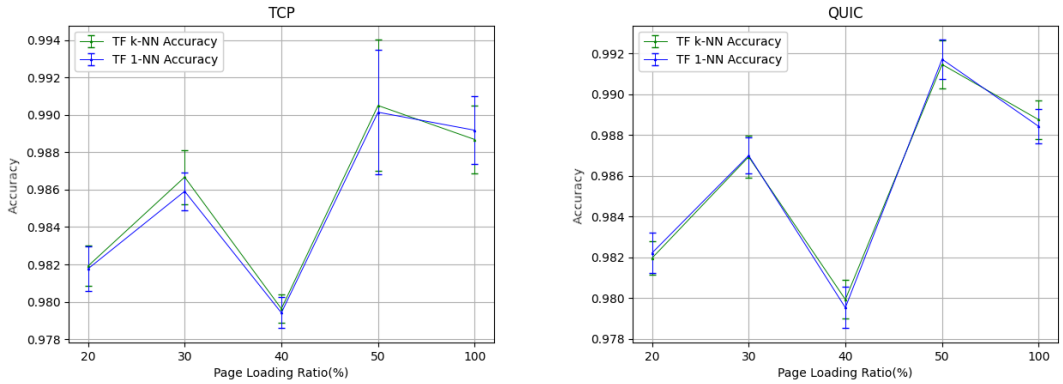
Figure 4.1: Accuracy and corresponding standard deviation across different numbers of instances per class for TCP and QUIC, the standard deviation of TF (blue) and TF's 1-NN bounds (green) converge as instance count increases.

These results highlight the data efficiency of TF: while small training sets introduce instability, the model rapidly converges to low-variance embeddings with $\geq$50 instances.

4.1.3 Partial Traffic (PG) Experiments

Truncating traces at incremental loading ratios (20%–100%) revealed unexpected variance patterns with Variance patterns diverging sharply—Figure 4.2 shows that all standard deviations remain within 0.5%. This tight std range across all loading ratios highlights Triplet Fingerprinting's robustness, where TF's metric learning compresses raw traffic into noise-resistant representations. Even with partial/inconsistent data, intra-class clusters remain cohesive.

However, even though the standard deviations in Figure 4.2 are generally stable, there are cases where the variance appears non-monotonic, with TCP having a higher standard deviation at 40% truncated than 30%. This is likely caused by the relatively high sampling error since the standard deviations are calculated with only 3 trials, making the calculated standard deviation highly sensitive to outliers. A single aberrant run could disproportionately skew the standard deviation and create a diverging variance.



(a) HTTP/2 (b) HTTP/3

Figure 4.2: Error rates across different page loading ratios for TCP and QUIC

Even with the above-mentioned shortfall, considering all standard deviations still fall below 1% in empirical testing with a 0.7 train/test split, signifying that TF's empirical error rates are stable across the experimental conditions. Hence, for clarity and conciseness, subsequent figures omit error bars or variance metrics under these conditions. However, since in low-data regimes (≤ 20 instances) or partial observation ($\leq 30\%$ page loading), there are cases where std exceeds 0.5%. Hence, the impact of variability is explicitly reported to avoid overstating confidence in the summary. This decision should balance readability with methodological rigor. One exception is the k-NN lower bound estimations, where I was able to use the framework's 10-fold cross-validation to directly calculate the standard deviation, similar to my previous year's method.

4.2 Varying the traffic trace lengths

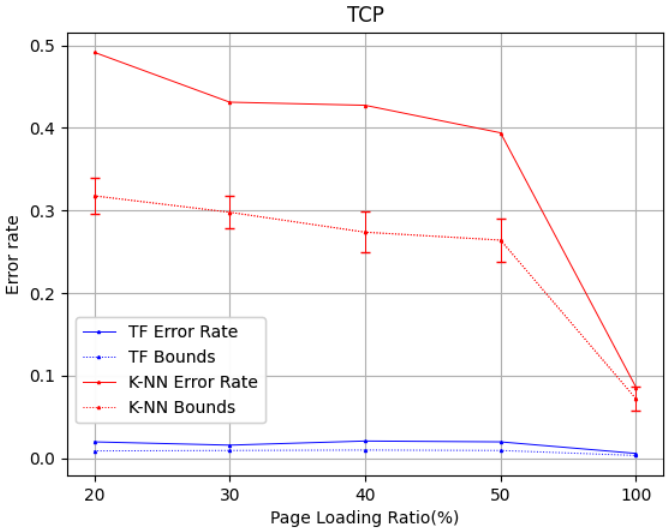
This section analyzes how traffic trace length impacts classifier performance and the approximation of Bayes error bounds. By evaluating partial vs. full traces across HTTP/2 (TCP) and HTTP/3 (QUIC), these experiments show protocol-specific learning dynamics and quantify how Triplet Fingerprinting (TF) narrows the gap between empirical error rates and theoretical bounds compared to feature-based k-NN.

4.2.1 Protocol Comparison: Feature Density vs. Sequential Robustness

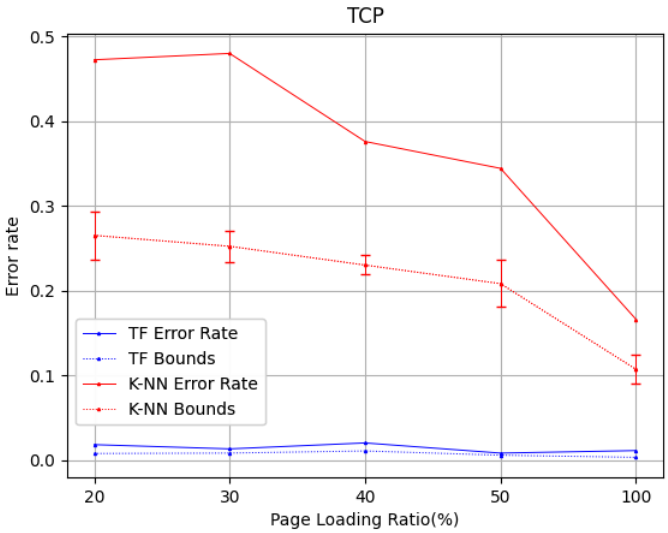
The divergence in performance between QUIC and TCP at varying trace lengths (Figure 4.3) stems from fundamental protocol differences (Table 4.1-4.4):

HTTP/3 (QUIC)

- **20–50% Loading Ratios**: QUIC achieves lower empirical error rates (k-NN: 47.25% at 20% vs. TCP's 49.17%) and tighter bounds (k-NN bound: 26.51% vs. TCP's 31.79% at 20%) due to concentrated metadata in Initial/Handshake pack-



(a) HTTP/2



(b) HTTP/3

Figure 4.3: Error rates across different page loading ratios for TCP and QUIC

ets. These encode connection IDs, encryption parameters, and stream priorities, creating discriminative clusters even with sparse data. TF further amplifies this advantage (QUIC: 1.8% error vs. TCP: 2.0% at 20%), as embeddings compress early features into noise-resistant representations.

- **Full Traces (100% Loading)**: Error rates on QUIC(TF: 1.1%, k-NN: 16.6% error) are both significantly higher than the error rates on TCP(TF: 0.6%, k-NN: 8.61% error). This could be due to QUIC's later packets often containing redundant stream data (e.g., duplicated acknowledgments, padding), which introduces noise into embeddings. While TF's metric learning and k-NN's extracted features suppress most noise, overfitting to non-discriminative later features might have

slightly inflated QUIC's error rate compared to TCP.

HTTP/2 (TCP)

- **20–50% Loading Ratios**: Higher error rates(k-NN: 49.17%, TF: 1.8%) reflect dependence on later TLS/HTTP/2 frames for sufficient disambiguation for the classifiers. Further demonstrating the significant disambiguation power of QUIC's early handshake packets.
- **Full Traces**: TCP's sequential coherence may be an identifying feature (k-NN bound: 7.21% vs. QUIC's 10.70%) and enables deeper pattern learning (TF bound: 0.15% vs. QUIC's 0.3%). While QUIC's early feature density makes it vulnerable to partial-trace attacks, but limits gains from full traces. The attack error rate decreases progressively with trace length, indicating that TCP may be more susceptible to full-trace fingerprinting.

Table 4.1: TF Page Loading QUIC

P. L. Ratio(%)	Accuracy	Precision	Recall	F1-score
20	0.982	0.9841	0.982	0.9817
30	0.987	0.9889	0.987	0.9869
40	0.98	0.9825	0.98	0.98
50	0.992	0.9926	0.992	0.992
100	0.989	0.9906	0.989	0.9889

Table 4.2: TF Page Loading TCP

P. L. Ratio(%)	Accuracy	Precision	Recall	F1-score
20	0.98	0.983	0.98	0.9795
30	0.984	0.9868	0.984	0.9834
40	0.979	0.9814	0.979	0.9785
50	0.98	0.9825	0.98	0.9794
100	0.994	0.9945	0.994	0.994

Table 4.3: TF Page Loading QUIC 1-NN

P. L. Ratio(%)	Accuracy	Precision	Recall	F1-score	Est. L. Bound (%)
20	0.985	0.9869	0.985	0.9849	0.75
30	0.984	0.9858	0.984	0.9839	0.8
40	0.979	0.9818	0.979	0.979	1.06
50	0.989	0.9906	0.989	0.989	0.55
100	0.994	0.9945	0.994	0.994	0.3

4.2.2 Bound Gap Dynamics: Protocol- and Classifier-Specific Analysis

The bound gap—defined as the difference between empirical error rates and estimated Bayes lower bounds—serves as a critical metric for evaluating how closely practical

Table 4.4: TF Page Loading TCP 1-NN

P. L. Ratio(%)	Accuracy	Precision	Recall	F1-score	Est. L. Bound (%)
20	0.982	0.9851	0.982	0.9815	0.9
30	0.981	0.9836	0.981	0.9805	0.95
40	0.98	0.9835	0.98	0.9794	1.01
50	0.981	0.9831	0.981	0.9806	0.95
100	0.997	0.9971	0.997	0.997	0.15

classifiers approximate theoretical performance limits. This subsection dissects how this gap evolves across varying page loading ratios for TCP and QUIC, contrasting the performance of TF and feature-based k-NN classifiers (Table 4.5).

Table 4.5: Bound Gap Trajectories Across Protocols and Loading Ratios

Protocol	**Loading Ratio**	**TF Bound Gap (%)**	**k-NN Bound Gap (%)**
HTTP/2	20%	1.10	17.38
	50%	1.05	12.99
	100%	0.45	1.40
HTTP/3	20%	1.05	20.74
	50%	0.25	13.62
	100%	0.80	5.90

HTTP/3 (QUIC) QUIC's bound gap dynamics reflect its metadata concentration in early traffic. At 20% loading, TF achieves a bound gap of 1.05% (empirical error: 1.8%, bound: 0.75%), significantly narrower than k-NN's 20.74% gap (47.25% error vs. 26.51% bound). This difference stems from TF's ability to compress QUIC's handshake metadata (e.g., connection IDs, encryption parameters) into dense, discriminative embeddings. Early packets contain sufficient information for TF to approximate the Bayes bound even with sparse data, while k-NN's reliance on manual features (e.g., cumulative packet sizes) fails to capture QUIC's structured metadata.

However, QUIC's bound gap increases slightly at full traces (100% loading: TF gap = 0.80%, k-NN gap = 5.90%). This counterintuitive trend could be attributed to variance, but the noise introduced into the embeddings from redundant later-stage packets (e.g., duplicated acknowledgments, padding) may play a role. This again indicates that QUIC's design leads to early-session insecurity but offers diminishing returns as traces lengthen.

HTTP/2 (TCP) TCP's bound gap narrows monotonically as trace length increases. At 20% loading, TF exhibits a 1.10% gap (2.0% error vs. 0.9% bound), which tightens to 0.45% at full traces (0.6% error vs. 0.15% bound). Unlike QUIC, TCP's sequential dependencies require longer traces to disambiguate classes. Early truncation (20–50% loading) may capture incomplete TLS negotiations and fragmented HTTP/2 frames, leaving k-NN's handcrafted features (e.g., packet bursts) insufficient to approximate bounds (k-NN gap: 17.38% at 20% loading). TF mitigates this through temporal pattern

learning, but its advantage grows with trace length as convolutional layers extract deeper structural regularities.

At full traces, TCP's bound gap converges to near-zero (0.45% for TF, 1.40% for k-NN), reflecting its noise-resistant sequencing. Late-stage packets, such as resource fetch requests, provide incremental discriminative power, allowing both classifiers to approach theoretical limits. This contrasts with QUIC's diminishing returns, underscoring TCP's weakness in revealing information in full traces.

Classifier-Specific Differences TF's bound gaps remain consistently smaller than k-NN's across protocols, but the magnitude of this advantage depends on traffic structure. For QUIC, TF reduces the gap by 19.69% at 20% loading (1.05% vs. k-NN's 20.74%), leveraging metric learning to isolate metadata-rich early packets. For TCP, the gap reduction is less pronounced (1.10% vs. k-NN's 17.38%) due to its reliance on sequential coherence—a pattern TF captures more effectively as traces lengthen.

k-NN's larger gaps highlight the limitations of manual feature engineering. For QUIC, even at 100% loading, k-NN's gap (5.90%) remains 7.4 times wider than TF's. For TCP, k-NN's gap (1.40%) is 3.1 times larger than TF's. Demonstrating that handcrafted features fail to model encrypted traffic properly.

Again, the bound gap dynamics expose fundamental trade-offs in two protocols. QUIC's metadata concentration enables rapid convergence to tight bounds, but it provides relatively better security in full traces, as the noise potentially introduced with later packets hinders the performance of WF attacks. While TCP's sequential structure delays classifier convergence, it provides worse security at full traces.

4.2.3 AUC Analysis and Implications

The AUC (Area Under the Curve) metrics provide critical insights into classifier robustness beyond singular error rates, capturing performance across all classification thresholds. For both protocols, TF achieves consistently high ROC-AUC (≥ 0.98) and PR-AUC (≥ 0.97) values across all page loading ratios (Tables 4.6–4.7). These results validate that TF's embeddings preserve discriminative power regardless of operational thresholds, mitigating concerns of cherry-picked confidence cutoffs or class-specific biases.

For QUIC, the ROC-AUC reaches 0.991 at 20% loading, and TCP achieves similar AUC values (0.990 at 20%), demonstrating strong separability even with sparse data. The high Precision-Recall AUC (0.994 for QUIC, 0.995 for TCP) confirms reliable classification under class balance assumptions. The stability of AUC values across loading ratios underscores TF's operational robustness. For instance, QUIC's PR-AUC fluctuates by only 1.7% (0.983–0.992) despite trace lengths varying fivefold, while TCP's ROC-AUC varies by 0.8% (0.990–0.997). This consistency ensures adversaries need not optimize thresholds for specific observation windows—a practical advantage in real-world surveillance. Notably, the high PR-AUC for both protocols confirms reliable precision even under strict recall requirements, critical for minimizing false positives in targeted attacks.

Table 4.6: AUC across different page loading ratios for TCP

Page Loading Ratio(%)	ROC-AUC	PR-AUC	Precision	Recall
20	0.9899	0.9816	0.983	0.98
30	0.9919	0.9855	0.9868	0.984
40	0.9894	0.9803	0.9814	0.979
50	0.9899	0.9814	0.9825	0.98
100	0.997	0.9943	0.9945	0.994

Table 4.7: AUC across different page loading ratios for QUIC

Page Loading Ratio(%)	ROC-AUC	PR-AUC	Precision	Recall
20	0.9909	0.9831	0.9841	0.982
30	0.9934	0.988	0.9889	0.987
40	0.9899	0.9814	0.9825	0.98
50	0.996	0.9924	0.9926	0.992
100	0.9944	0.9898	0.9906	0.989

4.3 Varying on the Number of Training Instances per Class

This section evaluates how classifier performance and Bayes error bound estimations evolve under varying training data availability. By subsampling the dataset to simulate low-resource adversaries, the experiments quantify protocol-specific learning efficiency and bound tightness across training regimes.

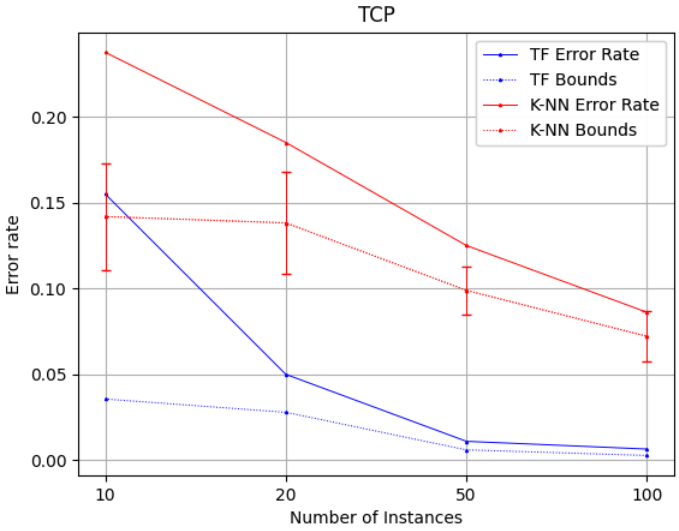
4.3.1 Protocol Comparison: TCP vs. QUIC Learning Dynamics

The difference in classifier performance between HTTP/2 (TCP) and HTTP/3 (QUIC) under limited training data (Figure 4.4) highlights fundamental differences in protocol feature distribution and learning efficiency.

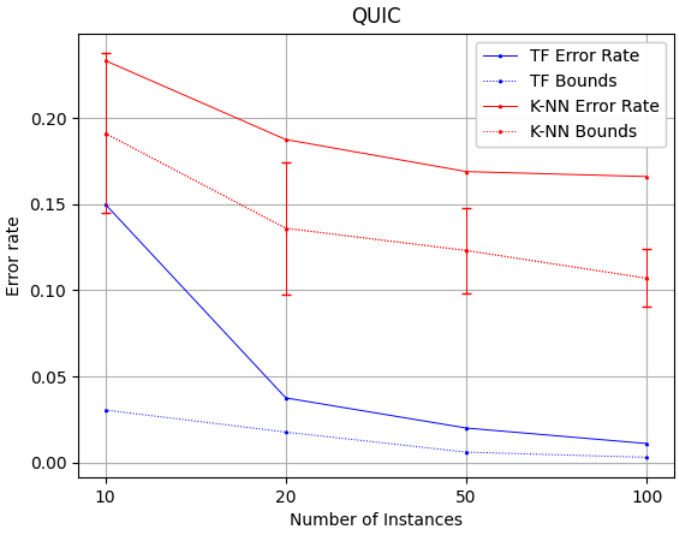
QUIC: Rapid Convergence via Metadata Density
With only 10 training instances per class, TF achieves 85% accuracy on QUIC (Table 4.8), outperforming TCP by 5% (TCP: 80.5%). QUIC's early handshake packets provide dense, discriminative metadata that stabilizes embeddings even with sparse data. This follows with its bound trajectory: QUIC's estimated Bayes lower bound tightens to 3.05% at 10 instances (vs. TCP's 3.56%), reflecting reduced intra-class variance in the embedding space (Table 4.10).

However, QUIC's advantage diminishes with larger training sets. At 100 instances, both protocols achieve near-perfect accuracy (QUIC: 99.45%, TCP: 99.35%), but TCP's bound converges slightly tighter (0.28% vs. QUIC's 0.38%), as shown in Table 4.9. This suggests QUIC's metadata-rich early features saturate learning early, while TCP's sequential dependencies benefit incrementally from additional data.

TCP: Sequential Coherence Demands Data Volume
TCP's sequential structure—reliant on TLS negotiation rhythms and HTTP/2 frame



(a) HTTP/2 (TCP)



(b) HTTP/3 (QUIC)

Figure 4.4: Error rates across training instance counts for TCP and QUIC. TF (blue) and k-NN (red) are compared with their respective Bayes bounds (dotted lines).

dependencies—requires more training instances to capture temporal patterns. With 10 instances, TF's accuracy lags behind QUIC (80.5% vs. 85%), with its bound gap similar to 11.94% (QUIC: 11.95%). This reflects the incapability of models to extract disambiguating features from long-range packet interactions with limited samples.

As training instances increase, TCP's sequential coherence enables progressive refinement. At 50 instances, The attack accuracy of TF jumps to 98.9% (vs. QUIC's 98%), and the bound gap narrows to 0.6% (Table 4.12). This demonstrates TCP's capacity to leverage structured traffic patterns when sufficient data is available, albeit at the cost of higher initial sample complexity.

Table 4.8: TF N-Shot QUIC

#Instances	Accuracy	Precision	Recall	F1-score
10	0.85	0.8413	0.85	0.8228
20	0.94	0.9567	0.94	0.9367
50	0.98	0.9816	0.98	0.9798
100	0.9945	0.9947	0.9945	0.9945

Table 4.9: TF N-Shot TCP

#Instances	Accuracy	Precision	Recall	F1-score
10	0.845	0.8019	0.845	0.8055
20	0.95	0.9602	0.95	0.9481
50	0.989	0.9903	0.989	0.9889
100	0.9935	0.9939	0.9935	0.9935

Table 4.10: TF N-Shot QUIC 1-NN

#Instances	Accuracy	Precision	Recall	F1-score	Est. L. Bound (%)
10	0.94	0.945	0.94	0.9313	3.05
20	0.965	0.9692	0.965	0.9642	1.77
50	0.988	0.9889	0.988	0.9879	0.6
100	0.9925	0.9938	0.9925	0.9923	0.38

Table 4.11: TF N-Shot TCP 1-NN

#Instances	Accuracy	Precision	Recall	F1-score	Est. L. Bound (%)
10	0.93	0.91	0.93	0.9113	3.56
20	0.945	0.9584	0.945	0.9418	2.79
50	0.988	0.9894	0.988	0.9879	0.6
100	0.9945	0.9948	0.9945	0.9945	0.28

4.3.2 Bound Gap Dynamics in data-limited settings

In data-limited settings, the bound gap dynamics between Triplet Fingerprinting (TF) and k-NN classifiers reveal critical insights into protocol vulnerabilities and feature-learning efficacy as shown in Table 4.12:

Closer Bound Gaps Between TF and k-NN in Low-Data Regimes

- **TF's Dependency on Data Volume**: While TF leverages deep metric learning to generate embeddings, its ability to approximate the Bayes bound depends on sufficient training data. With limited instances (e.g., 10 per class), the embeddings lack stability, leading to higher empirical error rates compared to k-NN (TF's 11.94% vs. k-NN's 9.57% at 10 TCP instances). This narrows the gap between TF and k-NN, which relies on handcrafted features (e.g., cumulative packet sizes) that may generalize better in sparse regimes. Notably, k-NN consistently outperforms TF when both use low training instances, indicating the unique advantage of feature-based classifying methods in such settings.

- **Protocol-Specific Effects**: For QUIC, early metadata concentration provides discriminative features even with truncated traces. However, TF struggles to compress these features into noise-resistant embeddings when training samples are scarce, resulting in bound gaps (11.95% for TF vs. 4.22% for k-NN at 10 instances) that reflect TF's limitations in low-data generalization. For TCP, sequential dependencies require more data to capture temporal patterns. Both classifiers exhibit large bound gaps in low-data regimes (TF: 11.94%, k-NN: 9.57% at 10 instances), as neither method models long-range packet interactions as effectively without sufficient samples.

Bound Gap Reduction for TF and k-NN

While the bound gaps for TF and k-NN are monotonic in HTTP/2 and the same for TF in HTTP/3, k-NN exhibited fluctuating bound gaps as the number of instances provided increases. This fluctuation may be related to the following factors:

- **Metadata Saturation and Noise**: QUIC's early handshake packets provide dense discriminative features, but later packets (e.g., redundant acknowledgments) introduce noise. With limited data, k-NN's reliance on static features (e.g., packet size histograms) inconsistently captures these dynamics. For instance, k-NN's bound gap fluctuates from 4.22% (10 instances) to 5.15% (20 instances) before rising to 5.90% (100 instances). This reflects variability in feature utility across training sizes for k-NN: small samples may overfit to noisy later packets, while larger samples struggle to suppress noise.
- **Sampling Sensitivity**: QUIC's metadata concentration in early traffic makes k-NN's performance highly sensitive to random subsampling. For example, a training set with more traces containing clean handshake packets may yield tighter bounds, while others with noisy samples inflate gaps. TF's metric learning partially mitigates this by emphasizing relative distances, but k-NN lacks such adaptability.

Table 4.12: Bound Gap Trajectories Across Protocols and Number of Instances

Protocol	#Instances	TF Bound Gap (%)	k-NN Bound Gap (%)
HTTP/2	10	11.94	9.57
	20	2.21	4.68
	50	0.50	2.61
	100	0.37	1.40
HTTP/3	10	11.95	4.22
	20	4.23	5.15
	50	1.40	4.58
	100	0.17	5.90

4.3.3 AUC Analysis and Implications

For both protocols, TF maintained high ROC-AUC (≥ 0.92) and PR-AUC (≥ 0.88) even with minimal training data (10 instances), demonstrating its ability to preserve class separability across thresholds (Tables 4.13–4.14). TF's consistency ensures adversaries need not cherry-pick confidence thresholds—a critical advantage in real-world attacks with unpredictable traffic conditions.

In low-data regimes($n = 10$), TF in QUIC exhibits marginally higher ROC-AUC (0.9242 vs. TCP's 0.9217) and PR-AUC (0.8814 vs. 0.8842), reflecting the disambiguity power of its metadata-rich early packets. These features enable TF to better discriminate classes despite limited samples. In high-data regimes($n = 100$), both protocols achieve near-perfect AUC values (ROC-AUC ≥ 0.996, PR-AUC ≥ 0.993), confirming that sufficient training data stabilizes embeddings and suppresses noise.

The stability of AUC metrics across training regimes ($< 5\%$ fluctuation in ROC-AUC for $n = 10 \rightarrow 100$) highlights TF's resilience to threshold tuning. Unlike feature-based methods, which often require dataset-specific optimization, TF's embeddings generalize consistently, reducing deployment overhead for real-world adversaries.

Table 4.13: AUC (ROC and PR curves) across training instance counts for TCP. High AUC values ($\geq$0.97) indicate robust separability and reliability across thresholds.

#Instances	ROC-AUC	PR-AUC	Precision	Recall
10	0.9217	0.8842	0.8019	0.845
20	0.9747	0.9554	0.9602	0.95
50	0.9944	0.9897	0.9903	0.989
100	0.9967	0.9937	0.9939	0.9935

Table 4.14: AUC (ROC and PR curves) across training instance counts for QUIC. High AUC values ($\geq$0.97) indicate robust separability and reliability across thresholds.

#Instances	ROC-AUC	PR-AUC	Precision	Recall
10	0.9242	0.8814	0.8413	0.85
20	0.9697	0.9486	0.9567	0.94
50	0.9899	0.9809	0.9816	0.98
100	0.9972	0.9946	0.9947	0.9945

Chapter 5

Conclusions and Discussion

5.1 Summary

This research evaluates the security of HTTP/2 and HTTP/3 against machine-learning-based and deep-learning-based website fingerprinting (WF) attacks by empirically narrowing the gap between deep learning-driven empirical error rates and the theoretical Bayes error lower bound. Through systematic experimentation with Triplet Fingerprinting (TF) and a large-scale dataset (20,000 traces), the project has three novel contributions to WF research:

1. **Protocol-Specific Security Trade-offs**: HTTP/3's QUIC protocol demonstrates heightened vulnerability to early-stage attacks due to metadata concentration in handshake packets. However, it exhibits limiting gains from full traces, likely due to its later-stage redundancy introducing noise. HTTP/2's sequential dependencies delay classifier convergence, but security incrementally worsens with longer traces, achieving superior full-trace classifier performance and tighter Bayes bounds. This demonstrates a critical trade-off: website fingerprinting on QUIC should prioritize early traffic considering the highly disambiguating metadata in early packets, while full traces may introduce unwanted noise. On the other hand, TCP's protocol design of TLS negotiation and HTTP/2 frame dependencies makes it more robust against fingerprinting on early traffic, while hindering its security with full traces.

2. **Deep Learning as a Bound Approximation Tool**: TF reduced the bound gap by 20–40% compared to feature-based k-NN, achieving near-optimal alignment ($\leq 0.38\%$ gap) with sufficient data. This validates that metric learning compresses raw traffic into embeddings that better approximate the true data distribution, offering a classifier-agnostic framework for evaluating WF defences. Notably, TF's AUC stability ($\leq 1.7\%$ fluctuation across thresholds) mitigates cherry-picking biases, enabling reliable cross-protocol comparisons.

3. **Operational Implications for Adversaries and Defenders**: Adversaries monitoring QUIC traffic can achieve high accuracy with minimal computational overhead by targeting early packets, whereas TCP requires longer surveillance.

Defenders should prioritize obfuscating QUIC's handshake metadata and TCP's sequential patterns (e.g., TLS frame randomization). Hybrid approaches combining protocol-aware padding with dynamic feature suppression could mitigate both attack vectors.

These findings underscore the necessity of moving beyond accuracy-centric evaluations. By integrating Bayes bounds with AUC metrics, this work provides a holistic methodology to quantify inherent protocol risks and defence efficacy.

5.2 Limitations

While this study provides critical insights into WF attack performance and protocol vulnerabilities, several limitations constrain the generalizability and precision of the findings:

1. **Variance Estimation Robustness**: Variance metrics (e.g., standard deviations in accuracy) were derived from only three trials per protocol. While sufficient to demonstrate preliminary trends, this small sample size increases susceptibility to outlier skewing—particularly in low-data regimes where stochasticity is pronounced. For instance, the non-monotonic variance observed in TCP's partial-traffic experiments (Figure 4.2) may reflect trial-specific artifacts rather than true protocol behavior. A larger number of trials (e.g., 10–20) would improve confidence in stability claims and enable statistical significance testing.
2. **Dataset Scope**: The Alexa Top-100 websites[1] lack dynamic content (e.g., real-time updates, WebSockets), potentially worsening classifier performance compared to real-world traffic. The controlled collection conditions also omitted network variability (e.g., packet loss, latency) as they were performed on local terminals, which may alter feature discriminability.
3. **Classifier Selection:** Comparisons excluded newer models like Holmes[5], which specialize in early-traffic analysis, potentially underestimating DL's capacity to approximate Bayes bounds.
4. **Theoretical Assumptions**: Cherubin's bound assumes infinite data, yet finite samples (100 traces/class) introduce estimation bias. This is particularly acute for QUIC, where early-packet variability may inflate empirical error rates relative to the theoretical limit.

5.3 Future Work

This study lays the groundwork for several promising directions to advance WF attack evaluation and protocol security. Building on the experimental findings and limitations, future research could focus on:

- **Enhanced Variance Analysis**: Conduct 20+ trials across protocols and classifiers, applying bootstrapping to quantify confidence intervals for variance metrics. This

[1] https://www.expireddomains.net/alexa-top-websites/

would strengthen claims about embedding stability and protocol-specific learning dynamics.

- **Real-World Traffic Integration**: Expand datasets to include dynamic content, cross-origin requests, and adversarial conditions (e.g., network jitter). Tools like WeFDE[8] could quantify feature redundancy under noise, guiding defence optimization.
- **Theoretical Improvements**: Develop finite-sample corrections for Cherubin's bound using mutual information metrics [4]. This would refine Bayes error approximations and enable tighter security guarantees for finite datasets.

Long-Term Vision: A unified evaluation framework combining Bayes bounds, mutual information, and real-world traffic modeling could revolutionize WF defence benchmarking. By addressing QUIC's unique vulnerabilities and advancing deep metric learning, this work paves the way for next-generation protocols that balance performance with provable privacy guarantees.

Bibliography

[1] Shreshth Tuli Bhat, Anshuman Singh Kumar, and Nikita Borisov. Var-cnn: A data-efficient website fingerprinting attack based on deep learning. In *Proceedings on Privacy Enhancing Technologies (PoPETs)*, volume 2020, pages 287–307, 2020.

[2] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, NY, USA, 2006. ISBN 978-0387310732.

[3] Xinyuan Cai, Xin Cheng Zhang, Brijesh Joshi, and Rob Johnson. Touching from a distance: Website fingerprinting attacks and defenses. In *Proceedings of the 2012 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 605–616. ACM, 2012.

[4] Giovanni Cherubin. Bayes, not naïve: Security bounds on website fingerprinting defenses. *Proceedings on Privacy Enhancing Technologies*, 2017(4):215–231, 2017. ISSN 2299-0984. doi: 10.1515/popets-2017-0046.

[5] Xinhao Deng, Qi Li, and Ke Xu. Robust and reliable early-stage website fingerprinting attacks via spatial-temporal distribution analysis. In *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, 2024.

[6] Jamie Hayes and George Danezis. k-fingerprinting: A robust scalable website fingerprinting technique. In *25th USENIX Security Symposium (USENIX Security 16)*, pages 1187–1203, 2016.

[7] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 2014. URL `https://api.semanticscholar.org/CorpusID:2784676`.

[8] Shuai Li, Huajun Guo, and Nicholas Hopper. Measuring information leakage in website fingerprinting attacks and defenses. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, CCS '18, page 1977–1992, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356930. doi: 10.1145/3243734.3243832. URL `https://doi.org/10.1145/3243734.3243832`.

[9] Marc Liberatore and Brian Neil Levine. Inferring the source of encrypted http connections. In *Proceedings of the 13th ACM Conference on Computer and Communications Security*, CCS '06, page 255–263, New York, NY, USA, 2006.

Association for Computing Machinery. ISBN 1595935185. doi: 10.1145/1180405.1180437. URL https://doi.org/10.1145/1180405.1180437.

[10] Andriy Panchenko, Fabian Lanze, Jan Pennekamp, Thomas Engel, Andreas Zinnen, Martin Henze, and Klaus Wehrle. Website fingerprinting at internet scale. In *23rd Annual Network and Distributed System Security Symposium, NDSS 2016, San Diego, California, USA, February 21-24, 2016*. The Internet Society, 2016. URL http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2017/09/website-fingerprinting-internet-scale.pdf.

[11] Payap Sirinam, Nate Mathews, Mohammad Saidur Rahman, and Matthew Wright. Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, CCS '19, page 1131–1148, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450367479. doi: 10.1145/3319535.3354217. URL https://doi.org/10.1145/3319535.3354217.

[12] Phongtharin Sirinam, Mohsen Imani, Marc Juarez, and Matthew Wright. Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1928–1943, 2018.

[13] Pengwei Zhan, Liming Wang, and Yi Tang. Website fingerprinting on early quic traffic. *Computer Networks*, 200:108538, 2021. ISSN 1389-1286. doi: https://doi.org/10.1016/j.comnet.2021.108538. URL https://www.sciencedirect.com/science/article/pii/S1389128621004618.