Poster: Towards Model Drift Resistant Website Fingerprinting with Time-Series LLMs

Yuwen Cui, Kai Wei, Kehan Shen, Guangjing Wang
University of South Florida, Tampa, FL, USA
{ycui, kwei, kshen, guangjingwang}@usf.edu

***Abstract*—Website fingerprinting (WF) attacks enable adversaries to infer users' browsing activities on anonymity networks such as Tor. The state-of-the-art WF techniques based on deep learning have achieved high attack accuracy. However, many approaches extract features from fixed-time windows and cache disabled browsers, leaving them insufficiently evaluated and often brittle under data drift. Additionally, website content and domain are updated frequently, which causes the concept drifting issue where the relationship between traffic features and traffic labels changes. Data drift and concept drift jointly constitute a difficult yet consequential form of model drift. To investigate the model drift, we collected Tor traffic over three months using a cache-enabled Tor Browser, covering 1,000 monitored websites with 500 traces per website, as well as 200,000 unmonitored webpage traces. With the dataset, we explore Time-series Language Models (TSLMs) to manage model drift efficiently and robustly. Specifically, we first construct an appropriate traffic representation for Tor. We then design a two-layer CNN to extract features from each traffic trace and generate corresponding traffic feature tokens as input to the TSLM. We expect more detailed system design and evaluation in the future.**

I. Introduction

The Tor network is a decentralized system that enables anonymous browsing using layered encryption and multi-hop relay routing. While designed to resist surveillance and traffic analysis, Tor traffic remains vulnerable to website fingerprinting (WF) attacks, which infer visited websites from traffic patterns. However, existing WF attacks [1], [2] against Tor traffic remain substantially challenged by model drift. In particular, frequent website updates and browser caching can alter traffic fingerprints over time, leading to degraded attack accuracy when models are deployed on data distributions that differ from those used during training.

The commonly adopted assumptions in WF either oversimplify real-world conditions or underestimate the defender's capabilities, thereby granting unrealistic advantages. First, many existing WF attack approaches [3] assume the static website content. However, modern websites are frequently changing content, such as ad-heavy or vlog platforms, or embedding third-party content, such as images, scripts, and videos. Though strategies are proposed to address frequent updates [4], the high variability of real-world traffic remains a major challenge. These rapid changes alter traffic patterns, making WF inefficiently adapt in dynamic environments.

Second, existing WF datasets [4] are typically collected with browser caching disabled. In this configuration, visiting a webpage triggers the download of all resource files, preventing access to the local cache. However, while the Tor Browser clears cached data between sessions to enhance privacy, it still uses cache files within a single browsing session. As a result, subsequent visits during the same session may load certain assets from the cache rather than re-downloading them, varying the observed traffic patterns.

To study how browser caching affects WF attack accuracy, we collected a dataset over three months, including over 1,000 monitored websites with 500 traces each and 200,000 unmonitored webpage traces. Our evaluation shows that state-of-the-art WF models suffer significant accuracy degradation on cache-enabled datasets. Based on these findings, we propose *WF-TSLM* for WF attacks under challenges from frequent website updates and browser caching. We design a two-layer CNN to extract features from traffic traces and generate feature tokens for the *WF-TSLM*. To improve pretraining, we introduce traffic direction transitions as fundamental patterns, helping the model learn stable and discriminative representations and reducing accuracy loss from network congestion. In this work, we evaluate an encoder-only LLM and plan to extend the analysis to decoder-only and encoder-decoder LLMs, such as GPT and LLaMA, in future work.

II. Methodology

A. Dataset Collection

Since existing WF datasets ignore the impact of an enabled browser cache, there is no definitive set of sensitive websites for WF research. Under this setting, we evaluate existing WF attack models using a browser cache-enabled dataset. We plan to release our dataset publicly in the future.

Specifically, we collected data from the 1,000 most popular websites, as ranked by Tranco [5], using Tor browser 14.0.8. For each website, we captured 500 network traces, including traces from subdomains, and removed duplicate websites, such as *google.com* and *google.ca*. To capture browser cache effects, we recorded an additional trace 15 seconds after the first visit. For the unmonitored dataset, we collected a single trace per page. In total, we gathered traces from the top 3,000 websites, excluding the top 1,000 monitored websites.

B. WF-TSLM Design

Our design extends the Time-LLM framework [6] with specialized components for Tor traffic classification. The architecture comprises six principal components: (1) packet direction pattern modeling, (2) CNN preprocessing with residual

TABLE I: Evaluation under browser cache disabled (AWF) and enabled (Ours) datasets.

Dataset	Attack Models (Acc)			
	AWF [4]	NetCLR [1]	*WF-TSLM*$_{\text{BERT}}$	*WF-TSLM*$_{\text{Longformer}}$
AWF Data	89.59%	89.80%	90.60%	92.57%
Ours	57.13%	63.45%	81.77%	85.28%

connections, (3) patch embedding, (4) LLM reprogramming, (5) bidirectional LLM processing, and (6) classification head with residual connections.

1) Packet Direction Pattern Module: To model packet direction transitions and burst patterns, we introduce a dedicated module that extracts and processes direction-specific features:

- **Transition Modeling**: Detects four types of direction transitions (outgoing→outgoing, outgoing→incoming, incoming→incoming, incoming→outgoing) using learned embeddings and multi-head attention.
- **Burst Pattern Analysis**: Tracks consecutive same-direction packets, recording burst lengths and positions (64 position bins) with dedicated embedding layers.
- **Feature Fusion**: Combines transition, burst, and statistical features through a two-layer Multi-Layer Perceptron (MLP) with LayerNorm and Gaussian Error Linear Unit (GELU) activation.

The output of this module provides explicit representations of packet direction patterns that complement the learned features from the LLM backbone.

2) CNN Preprocessing with Residual Connections: Before patch embedding, the packet direction sequence is processed through a two-layer CNN with residual connections designed to preserve directional patterns:

$$H_{\text{CNN}} = \text{CNN}_{\text{layer2}}(\text{CNN}_{\text{layer1}}(X)) + \text{Residual}(X). \quad (1)$$

The residual connection ensures that the original packet direction information ($-1/+1$) flows through the network, preventing information loss during feature extraction.

3. Patch Embedding: The CNN-processed sequence is divided into overlapping patches of length $p = 16$ with stride $s = 8$. Each patch is embedded into a d_{ff}-dimensional space:

$$E = \text{PatchEmbedding}(H_{\text{CNN}}; p, s), \quad (2)$$

where $E \in \mathbb{R}^{P \times d_{\text{ff}}}$ and $P = \lfloor (L-p)/s \rfloor + 1$ is the number of patches.

4. LLM Reprogramming: The patch embeddings are reprogrammed to align with the pre-trained LLM's input space using a cross-attention mechanism:

$$E_{\text{reprog}} = \text{ReprogrammingLayer}(E, W_{\text{LLM}}, W_{\text{LLM}}) \quad (3)$$

where W_{LLM} is the pre-trained LLM's embedding matrix.

III. Evaluation

In this section, we investigate the accuracy degradation caused by the model drift in existing WF attack models using our collected browser cache-enabled dataset and the existing AWF dataset [4]. The expanded AWF dataset was collected over six weeks to capture website updates, which is widely regarded as a benchmark for studying concept drift in WF attacks. To better understand how TSLMs capture and interpret Tor traffic patterns under these two conditions, we conduct a comparative study of representative encoder-only LLMs, including *WF-TSLM*$_{\text{BERT}}$ and *WF-TSLM*$_{\text{Longformer}}$.

TABLE II: Concept drift evaluation on the AWF dataset under website changes after 3 days, 2 weeks, 4 weeks, and 6 weeks.

Models	3 days (Acc)	2 weeks (Acc)	4 weeks (Acc)	6 weeks (Acc)
AWF$_{\text{CNN}}$	89.59%	89.13%	86.29%	81.35%
WF-TSLM$_{\text{BERT}}$	90.60%	90.60%	90.13%	89.21%
WF-TSLM$_{\text{Longformer}}$	92.57%	92.55%	91.89%	90.80%

TABLE I and TABLE II present a performance comparison between representative encoder-only LLM-based models and two state-of-the-art WF attacks. The results show that existing WF attack models suffer from significant accuracy degradation when browser caching is enabled. In contrast, our proposed TSLM model improves the classification accuracy by up to 28.15% compared to AWF. Moreover, the concept drift evaluation demonstrates that our model maintains strong robustness over time: after six weeks of website updates, the accuracy drops by less than 2.0%, which is substantially lower than the approximately 8.0% degradation observed for AWF.

IV. Conclusion

In this work, we investigate the accuracy degradation caused by model drift in existing WF attack models. We first introduce a dataset collected with the browser cache enabled setting. We then evaluate model drift in WF attack models using both the collected dataset and the existing AWF dataset. To address the observed accuracy degradation, we propose *WF-TSLM*, a robust and efficient WF attack model based on the TSLM framework. To evaluate the performance of *WF-TSLM*, we conduct comprehensive experiments on both our dataset and the AWF datasets. The results demonstrate that the TSLM model effectively mitigates the model drift problem.

References

[1] A. Bahramali, A. Bozorgi, and A. Houmansadr, "Realistic website fingerprinting by augmenting network traces," in *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, 2023, pp. 1035–1049.

[2] M. Shen, J. Wu, J. Ai, Q. Li, C. Ren, K. Xu, and L. Zhu, "Swallow: A transfer-robust website fingerprinting attack via consistent feature learning," in *Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security*, 2025, pp. 1574–1588.

[3] Y. Cui, G. Wang, K. Vu, K. Wei, K. Shen, Z. Jiang, X. Han, N. Wang, Z. Lu, and Y. Liu, "A comprehensive survey of website fingerprinting attacks and defenses in tor: Advances and open challenges," *arXiv preprint arXiv:2510.11804*, 2025.

[4] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," *arXiv preprint arXiv:1708.06376*, 2017.

[5] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczyński, and W. Joosen, "Tranco: A research-oriented top sites ranking hardened against manipulation," in *Proceedings of the 26th Annual Network and Distributed System Security Symposium*, ser. NDSS 2019, Feb. 2019.

[6] M. Jin, S. Wang, L. Ma, Z. Chu, J. Y. Zhang, X. Shi, P.-Y. Chen, Y. Liang, Y.-F. Li, S. Pan *et al.*, "Time-llm: Time series forecasting by reprogramming large language models," *arXiv preprint arXiv:2310.01728*, 2023.

Poster: Towards Model Drift Resistant Website Fingerprinting with Time-Series LLMs

Yuwen Cui, Kai Wei, Kehan Shen, Guangjing Wang

Bellini College of Artificial Intelligence, Cybersecurity and Computing

UNIVERSITY of SOUTH FLORIDA

Introduction

Tor routes traffic through three relays, entry, middle, and exit, rotated periodically and encrypted using onion routing so no single relay knows both the source and destination. Despite this design, Tor traffic remains vulnerable to traffic analysis, such as website fingerprinting (WF) attacks that infer visited websites from traffic patterns.

Motivation

1. Many WF approaches avoid dynamic websites, such as ad-heavy or vlog platforms. Third-party resources further blur traffic boundaries, causing traffic patterns to change and reducing the accuracy and generalization of WF attacks.

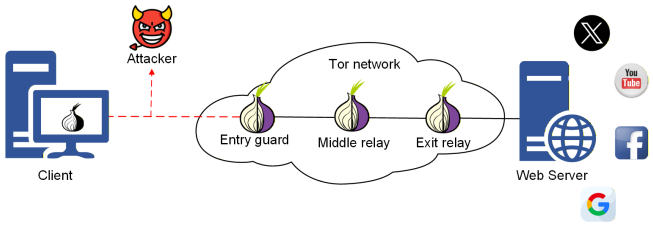
2. Website fingerprinting datasets are usually collected with caching disabled, producing identical traffic patterns across visits. However, Tor uses caching within a session, causing repeated visits to differ and introducing variability that can undermine experiments ignoring caching effects.

Contribution

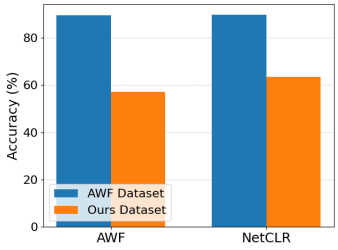
1. We collected a large cache-enabled dataset to study the impact of browser caching on WF accuracy and found that SOTA WF attack models suffer significant performance degradation.

2. We propose ***WF-TSLM***, the first TSLM-based approach for website fingerprinting, designed to handle dynamic websites and cache-enabled traffic using CNN-extracted traffic feature tokens.

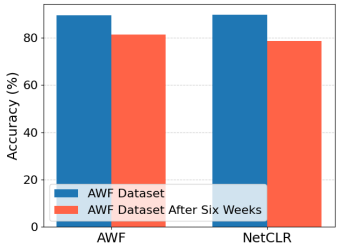
Overview of the WF attack in Tor architecture



Challenges of Existing WF Attack Models

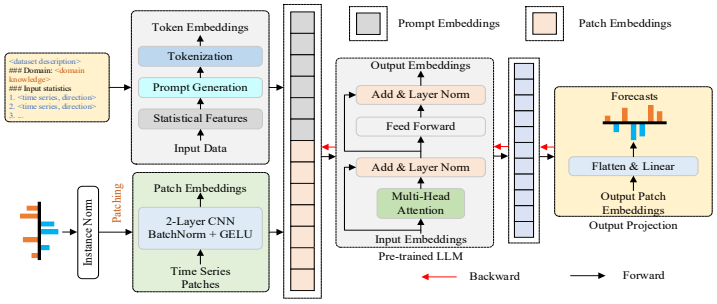


This figure shows the accuracy of the AWF and NetCLR attack models on the AWF dataset and our browser cache-enabled dataset. Both models exhibit significant accuracy degradation when the browser cache is enabled.



This figure shows the accuracy changes of the models over the six weeks of website updates from the AWF dataset. Both models exhibit accuracy degradation due to frequent website updates.

WF-TSLM Architecture



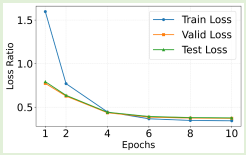
Evaluation

TABLE I: Evaluation under browser cache disabled (AWF) and enabled (Ours) datasets.

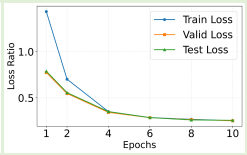
Dataset	Attack Models (Acc)			
	AWF [4]	NetCLR [1]	*WF-TSLM*$_{\text{BERT}}$	*WF-TSLM*$_{\text{Longformer}}$
AWF Data	89.59%	89.80%	90.60%	92.57%
Ours	57.13%	63.45%	81.77%	85.28%

TABLE II: Concept drift evaluation on the AWF dataset under website changes after 3 days, 2 weeks, 4 weeks, and 6 weeks.

Models	3 days (Acc)	2 weeks (Acc)	4 weeks (Acc)	6 weeks (Acc)
AWF$_{\text{CNN}}$	89.59%	89.13%	86.29%	81.35%
WF-TSLM$_{\text{BERT}}$	90.60%	90.60%	90.13%	89.21%
WF-TSLM$_{\text{Longformer}}$	92.57%	92.55%	91.89%	90.80%



(a) Loss ratio from *BERT* model.



(b) Loss ratio from *Longformer* model.

Figure (a) and (b) show the Train, Validation, and Test losses of the *WF-TSLM*$_{\text{BERT}}$ and *WF-TSLM*$_{\text{Longformer}}$ models on the AWF dataset over the first 10 epochs.

References

[1] A. Bahramali, A. Bozorgi, and A. Houmansadr, "Realistic website fingerprinting by augmenting network traces," in Proceedings of the 2023ACM SIGSAC Conference on Computer and Communications Security,2023.

[2] M. Shen, J. Wu, J. Ai, Q. Li, C. Ren, K. Xu, and L. Zhu, "Swallow:A transfer-robust website fingerprinting attack via consistent featurelearning," in Proceedings of the 2025 ACM SIGSAC Conference onComputer and Communications Security, 2025.

[3] Y. Cui, G. Wang, et. al., "A comprehensive survey of website fingerprinting attacks and defenses in tor: Advances and open challenges," arXiv preprint arXiv:2510.11804, 2025.

[4] V. Rimmer, D. Preuveneers, M. Juarez, T. Van Goethem, and W. Joosen, "Automated website fingerprinting through deep learning," arXiv preprintarXiv:1708.06376, 2017.

[5] V. Le Pochat, T. Van Goethem, S. Tajalizadehkhoob, M. Korczy´nski,and W. Joosen, "Tranco: A research-oriented top sites ranking hardenedagainst manipulation," in Proceedings of the 26th Annual Network andDistributed System Security Symposium, ser. NDSS 2019, Feb. 2019.