Hybrid CNN-SSM Model for Robust Multi-Tab Website Fingerprinting over Tor

Zulu Okonkwo*, Ernest Foo*, Zhe Hou*, Qinyi Li*, and Zahra Jadidi*
*Griffith University, Australia
{z.okonkwo, e.foo, z.hou, qinyi.li, z.jadidi}@griffith.edu.au

***Abstract*—Website fingerprinting (WF) threatens anonymity networks such as Tor by inferring visited sites from encrypted traffic. While recent deep learning attacks report high accuracy, they are often evaluated under simplifying assumptions: single-tab browsing, or multitab settings where the number of open tabs is known. In addition, most work treats WF as a pure classification problem and provides limited insight into what the model learns. We present a multitab WF attack that handles an *unknown* number of concurrently opened tabs and explicitly incorporates explainability into evaluation. Our pipeline segments traces, encodes packet, burst, and timing-level statistics that remain informative under common Tor defences, and learns representations with a dilated residual CNN for hierarchical local structure and a state-space model for long-range temporal dependencies. Experiments on public WF benchmarks, including defended traffic, show that our method outperforms prior state-of-the-art methods in multitab settings while remaining competitive in single-tab classification.**

***Index Terms*—tor, web fingerprinting, deep learning, mamba**

I. Introduction

Tor is a widely used anonymity service with over 2 million daily users and growing [22], making it a prominent target for website fingerprinting (WF), where adversaries infer visited websites from side-channel information such as traffic patterns and metadata. Recent WF attacks [9], [10], [13], [14], [24]–[28] use sophisticated DL models and often exceed 90% accuracy, motivating defences that pad traffic or introduce adaptive delays [6], [7], [15]–[17], [21], sometimes reducing accuracy below 10% [21]. However, DL-based WF has been criticized for relying on unrealistic assumptions, including access to complete, unmodified traces [9], [10], [13], [14], [24] despite noise, loss, and limited vantage points in practice, and for assuming single-tab browsing even though users commonly open multiple tabs or pages, producing interleaved traffic that can significantly degrade WF performance [27]. Finally, most WF work prioritizes accuracy over explainability, even though understanding model evidence is essential for analysing multitab behaviour and guiding stronger defences.

To address these limitations, we build on recent WF advances [24]–[26], [32] and propose a model that remains effective under diverse defences and improves over prior SOTA. Table I summarizes how existing attacks compare to our approach in multitab capability, robustness to common defences (^{a}), genericity to unknown tab counts (^{b}), and explainability (^{c}). Our analysis highlights two primary drivers of DL-based WF performance: feature extraction and representation learning, and motivates a representation that combines packet counts, burst structure, and burst-timing sequences to capture both fine local cues and more stable global patterns (coarse burst features are known to remain informative under distortion [24]). Architecturally, we adopt a hybrid design: a dilated residual CNN extracts hierarchical patterns with an expanded receptive field, and a state-space model (SSM) aggregates long-range dependencies across the trace. Compared with transformer-based backbones commonly used for long-context modeling [25], [26], SSMs process sequences in linear time without requiring full-context self-attention, making them a natural fit for long, interleaved multitab traffic. In this paper, we make the following contributions:

- We propose a generic multi-tab Tor WF attack that remains accurate under defences and unknown tab counts using an SSM-based classifier.
- We evaluate on public datasets across multiple settings, demonstrate superior performance, and release our code and experimental framework for reproducibility.
- We provide a thorough explanation of our model's behavior to ensure that the proposed method is transparent, trustworthy, and faithful to the decision process.

TABLE I
Comparison of methods across key properties.

Methods	Multi-tab	Robustness^a	Genericity^b	Explainability^c
DF [9]	✗	✓	✗	✗
Var-CNN [12]	✗	✓	✗	✗
Tik-Tok [14]	✗	✓	✗	✗
BAPM [31]	✓	✗	✗	✗
RF [24]	✗	✓	✗	✗
TMWF [26]	✓	✓	✗	✗
NetCLR [23]	✓	✓	✗	✗
ARES [32]	✓	✓	✓	✗
Ours	✓	✓	✓	✓

II. Background and Related Work

1) WF Attacks: Early WF work framed single-tab WF as supervised classification and relied on ML with engineered features capturing counts, ordering, and timing statistics [3]–[5], [8]. Wang et al. [3] used distance-weighted k-NN, Panchenko et al. [5] proposed CUMUL with cumulative-sum representations and an SVM, Hayes et al. [4] introduced k-FP using feature selection with random forests and k-NN matching, and Yan et al. [8] expanded feature engineering

with TCP/IP header-derived attributes evaluated using SVM, *k*-NN, and Extra Trees. As representation learning matured, WF shifted toward DL, with CNN-based architectures dominating due to automated extraction of discriminative local patterns from traces [9], [10], [12]–[14], [23]–[25], [29], [35]. AWF [10] showed DL can rival feature-based ML, DF [9] surpassed prior baselines and defeated WTF-PAD [6], and Var-CNN [12], Tik-Tok [14], and TF [13] further improved accuracy and data efficiency; GAN-based variants have also been explored [20].

More recent work targets multitab browsing, where concurrent page loads interleave traffic and weaken per-site signatures [1], [28], [30], [32], [34]. CountMamba [1] reconstructs TLS records and models timestamp sequences with signed record lengths using a causal CNN and a Mamba stack, but assumes access to TLS record length beyond the standard direction and timing only setting. ARES [32] formulates multitab WF as multi-label classification with transformer-based modelling, but requires per-website components and can degrade on long traces. FMWF [34] uses few-shot learning via synthetic multitab pre-training, which may not fully reflect real multitab interactions. Oscar [28] targets subpage identification via DF-style feature transformations and multi-label metric learning, and Mitseva et al. [30] show that modeling sets of subpages can further improve WF. Despite these advances, multitab WF remains challenging in practice, and explainability is rarely considered.

2) WF Defences: As DL based WF attacks demonstrated high effectiveness, researchers focused on developing countermeasures aimed at disrupting these attacks. Proposed defences primarily delay, split or pad traces. Delay-based approaches such as Tamaraw [2] inject timing perturbations to disrupt temporal consistency. Traffic-splitting defences, such as Traffic sliver [15], fragment traffic across multiple paths or sessions to conceal local features. Padding-based defences such as WTF-PAD [6], Walkie-Talkie [7], Front [16], RegularTor [17], Mockingbird [18], and Blanket [19], introduce dummy traffic or standardised burst sizes to obscure patterns but still introduce high traffic overhead.

3) Beyond Local: Coarse and Global Context: CNNs capture fine-grained local patterns, but padding defences such as Front [16] can suppress these cues and reduce attack effectiveness. Since traffic traces are sequential, incorporating broader context is often necessary under defence. TAM [24] shows this by segmenting traces and feeding coarse, segment-level summaries to a CNN. Hybrid models such as ARES [32] and TMWF [26] combine a CNN for local features with a transformer for long-range dependencies in multitab sessions, while Laserbeak [29] augments a DF-style backbone with attention modules to encode global relationships. Despite advances, these methods still struggle in multitab scenarios.

III. Method

A. Traffic Representation

Shen et al. [24] showed that counting packets in fixed time slots remains effective for single-tab WF even under Tor defences, whereas Deng et al. [32] found that packet-level features alone are insufficient for multitab WF and therefore incorporated burst-level features, extensive preprocessing, and per-website classifiers [32]. Building on these insights, we adopt a hybrid feature design that combines packet, burst, and timing information for multitab WF. Since traffic traces are sequential time series, our end-to-end model (Figure 2) takes a trace and preset hyperparameters and comprises three stages: trace processing, local feature extraction, and global modelling. **Trace Processing.** A website visit produces a

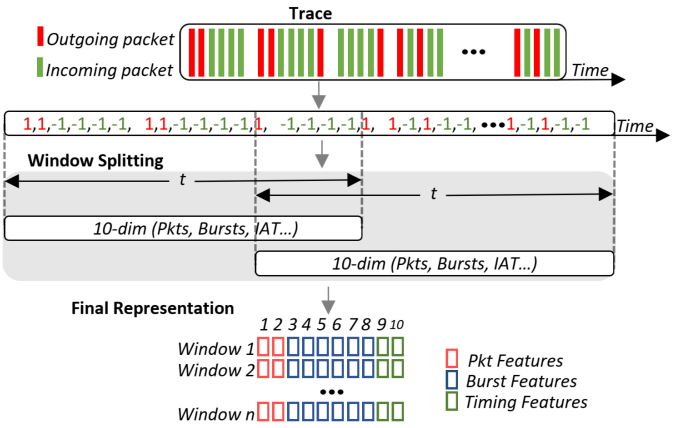

Fig. 1. Trace processing constructs a new representation by splitting each trace into overlapping windows and computing feature stacks for each segment.

trace $W = (w_1, \ldots, w_l)$, where each event $w_k = (d_k, t_k)$ contains packet direction $d_k \in \{+1, -1\}$ (outgoing/incoming) and timestamp t_k. As shown in Figure 1, we convert the trace to a signed sequence and segment it with a sliding window of 20 ms and 10 ms stride (50% overlap) to obtain temporally aligned subsets that are less sensitive to timing shifts (See Figure 1). We compute features per window and stack them into an $n \times 10$ matrix representation. We extract a fixed set of 10 features per window. At the packet level, we compute the outgoing and incoming packet counts. At the burst level, where a *burst* is a maximal run of consecutive packets in the same direction, we compute the number of outgoing bursts and incoming bursts, the median outgoing and incoming burst sizes, and the maximum and minimum burst lengths (across either direction). Finally, for timing, we compute the mean inter-arrival time (IAT) for outgoing packets and for incoming packets (in ms). Stacking these per-window features over all windows yields the feature matrix used for WF.

Hierarchical temporal modelling. CNNs are widely used for WF feature extraction [9], [10], [12]–[14], [23]–[25], [29], [32], [35] and remain the backbone of many SOTR classifiers because translation-equivariant filters detect local patterns regardless of position and deeper stacks with pooling become less sensitive to temporal shifts, improving tolerance to timing noise and misalignment in defended traces. To capture both short- and long-range structure in segmented traffic, we use a dilated residual 1D CNN: a stem layer (Conv1D + BatchNorm + ReLU) maps the 10-dimensional window

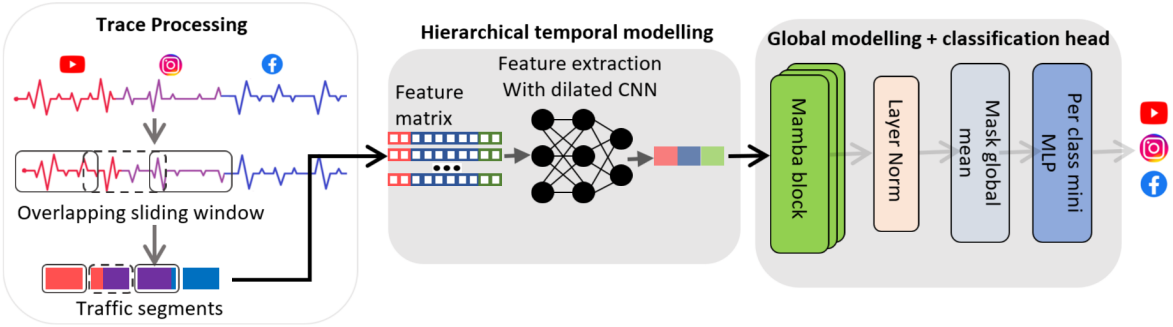

Fig. 2. Model Overview: The entire process has three distinct parts. Windowed feature extraction yields a structured sequence representation, which is first processed by a dilated convolutional encoder to capture short-range structure and subsequently by a state-space model to model global temporal dependencies.

features to a higher-dimensional space, followed by residual blocks with two Conv1D layers, BatchNorm, dropout, and skip connections, where dilation grows exponentially as $d_b = 2^b$ for $b \in \{0, \ldots, N\}$ to expand the receptive field without downsampling and preserve sequence resolution; finally, a 1×1 pointwise convolution projects CNN channels to the Mamba embedding dimension for the subsequent state-space module.

Global modelling and classification. Before the Mamba stack, we add absolute positional embeddings to enable the model to distinguish where events occur within a segment, and we apply LayerNorm. We then pass the sequence through three Mamba blocks, which can be viewed as learning continuous-time state dynamics with data-driven time constants that control how quickly information is integrated or forgotten, a useful inductive bias for multitab traces with entangled long-range dependencies. The Mamba stack aggregates cross-segment context while preserving the fine-scale temporal structure produced by the dilated CNN. We apply a final LayerNorm for stability and use masked global mean pooling to form a trace-level embedding, ignoring padding. The pooled embedding is fed to per-class mini-MLP heads (dropout, normalization + linear, SiLU, linear) trained with binary cross-entropy with logits, keeping the heads lightweight to reduce overfitting and keep decisions close to the learned global representation.

IV. Evaluation

A. Experimental Setup

1) Implementation: We generate the traffic representations, build, train, validate, and test our model using PyTorch 2.3.1 and Python 3.10.16 backend. The hardware specification is a Linux 6.2.0-26-generic server. The processor is a 12th-generation Intel(R) Core(TM) i9-12900, with 125GB of physical RAM and an NVIDIA RTX A4000 GPU. For all experiments, we split each dataset into three sets: 80% training, 10% validation and 10% testing.

2) Hyperparameters: The hyperparameters used throughout the training and validation process are as follows. Each traffic trace is segmented into overlapping windows of 20ms, with a stride of 10ms, and the resulting sequence is capped at a maximum length of 10,000. The Hierarchical temporal modelling stage comprises eight residual blocks, where the dilation factor grows exponentially with the block index. The n-th block uses a dilation of 2^n, yielding a maximum dilation of 2^7. A kernel of 3 is used, and the Rectified Linear Unit(ReLU) activation function is also utilised for all the CNNs in this module. The global modelling stage consists of three mamba blocks, followed by a per-class mini-MLP head that employs the Sigmoid Linear Unit (SiLU) activation. For model training, the batch size is 64, and the AdamW optimizer is used with a learning rate of 0.0003 and a weight decay of 0.01. The model is trained for 100 epochs, with early stopping after 10 epochs without improvement. Binary cross-entropy with logit loss is used for the final classification, enabling independent binary predictions for all output classes. Hyperparameter tuning was confined strictly to the training and validation sets. The test set is held out and not used during tuning, ensuring its integrity for an unbiased evaluation.

3) Datasets: We first validate our model in a single-tab setting to establish a clean baseline of its feature extraction performance. We then evaluate on the full multi-tab dataset to quantify how well this performance carries over to realistic conditions.

Single-tab Datasets. We report single-tab results only in the closed-world setting, since this work focuses on multitab WF. We use three common datasets: DF [9] (95 websites, 1000 undefended traces each), Walkie-Talkie [14] (900 Walkie-Talkie-protected traces over 100 websites), and Wang [7] (100 websites, 100 undefended traces each). To test performance under defences, we use the released Walkie-Talkie traces [7], simulate WTF-PAD [6], FRONT [16], TrafficSliver [15], and RegularTor [17] on DF, and generate Tamaraw [2] and CS-BuFLO [33] traces from Wang.

Multi-tab Datasets. We evaluate multitab WF on the ARES dataset [25], which collects traces from Alexa Top sites under realistic concurrent browsing with 2-5 tabs and provides both closed and open-world splits. In the closed world, the monitored set contains Alexa's top 100 websites, and each trace corresponds to a random combination of k monitored sites [25]. In the open world, combinations include $k-1$ monitored sites plus one non-monitored site sampled from Alexa's top 20,000 (excluding the top 100), with each non-monitored site used once [25]. Labels cover 100 monitored

TABLE II

CLOSED-WORLD SINGLE-TAB CLASSIFICATION ACCURACY (%) UNDER DIFFERENT DEFENCES. AVERAGE ACCURACY ACROSS DEFENCES IS SHOWN IN THE RIGHTMOST COLUMN. BEST RESULTS PER COLUMN ARE IN **BOLD** AND SECOND BEST RESULTS ARE UNDERLINED.

Attacks ↓	Undefended	WalkieTalkie	WTFPAD	Front	TrafficSliver	RegularTor	Tamaraw	CS - BuFLO	Avg. Defence
AWF [10]	94.32 ± 0.68	29.61 ± 0.63	52.67 ± 3.65	36.85 ± 2.69	3.86 ± 1.03	10.79 ± 1.10	7.06 ± 1.38	11.00 ± 0.01	21.69
TF [13]	97.88 ± 0.05	47.83 ± 0.21	85.30 ± 0.13	69.95 ± 0.50	46.58 ± 0.05	15.18 ± 1.12	6.00 ± 0.02	1.03 ± 0.02	38.84
NetCLR [23]	98.45 ± 0.00	43.96 ± 0.03	88.20 ± 0.02	80.77 ± 0.04	59.02 ± 0.05	22.66 ± 0.17	12.00 ± 0.03	12.40 ± 0.11	45.57
BAPM [31]	96.55 ± 0.05	41.79 ± 0.01	83.45 ± 0.03	65.98 ± 0.08	46.81 ± 0.08	23.14 ± 0.05	17.20 ± 0.05	**14.20** ± 0.05	41.80
DF [9]	98.40 ± 0.11	46.02 ± 0.89	90.86 ± 0.82	79.80 ± 0.56	39.45 ± 0.08	22.96 ± 1.43	8.75 ± 0.72	12.05 ± 0.49	42.84
Tik - Tok [14]	98.45 ± 0.13	72.85 ± 0.56	93.80 ± 0.47	84.79 ± 0.14	57.43 ± 4.45	22.86 ± 5.80	6.94 ± 0.18	13.33 ± 0.05	50.29
Var - CNN [12]	98.87 ± 0.05	87.53 ± 1.10	94.10 ± 0.31	79.24 ± 3.06	39.93 ± 0.05	15.37 ± 7.52	3.13 ± 1.31	3.71 ± 0.02	44.97
TMWF [26]	97.22 ± 0.11	46.48 ± 0.01	90.02 ± 0.01	80.33 ± 0.11	56.51 ± 0.05	**25.11** ± 0.10	**26.70** ± 0.07	**16.78** ± 0.09	48.85
RF [24]	98.84 ± 0.01	93.87 ± 0.23	**96.58** ± 0.13	**93.34** ± 0.18	79.68 ± 0.22	7.63 ± 1.10	3.30 ± 1.31	1.22 ± 0.02	53.66
Ours	98.07 ± 0.15	**98.72** ± 0.02	95.84 ± 0.05	92.93 ± 0.03	**84.05** ± 0.06	21.83 ± 1.01	7.60 ± 3.17	12.60 ± 2.02	**59.08**

classes and a single aggregated non-monitored class (101 classes total).

4) Baselines: Our study ensures a fair comparison by benchmarking against state-of-the-art deep learning based WF attacks. We evaluate our model against CNN-based attacks AWF [10], DF [9], Tik-Tok [14], Var-CNN [12], RF [24], BAPM [31], NetCLR [23], TMWF [26] and ARES [32]. We deploy the WF defences described in Section IV-A3 in the single tab setting. To deploy single-tab attacks (DF, Tik-Tok, Var-CNN, RF) in multitab settings and multitab attacks (BAPM, NetCLR, TMWF) in single-tab settings, we adapt the models by revising the loss and, when required, reconfiguring the output layer.

5) Metrics: We employ four metrics across six categories:

Single-tab Metrics: We report accuracy using 5-fold cross-validation: in each fold, three splits are used for training, one for validation, and one for testing, with the model re-initialized each time. Results are the mean and standard deviation over the five folds.

Multi-tab Metrics: In the multitab setting, we report AUC, $P@K$, and $\mathrm{MAP@}K$. AUC (area under the ROC curve) measures ranking quality of true sites above non-sites across all thresholds. $P@K$ measures how many of the top-K predicted websites are correct. For an instance with multi-label ground truth $\mathbf{y} \in \{0,1\}^C$ and predicted scores $\hat{\mathbf{y}} \in [0,1]^C$, let $r_K(\hat{\mathbf{y}})$ denote the indices of the top-K scores; then

$$P@K(x) \;=\; \frac{1}{K} \sum_{j \in r_K(\hat{\mathbf{y}})} y_j. \quad \in [0,1]. \tag{1}$$

$\mathrm{MAP@}K$ further accounts for the rank positions of correct websites within the top-K list:

$$\mathrm{MAP@}K \;=\; \frac{1}{K} \sum_{j=1}^{K} \mathrm{P@}j. \tag{2}$$

B. Single-Tab

Table II reports single-tab accuracy under selected Tor defences, with the best result in bold and the runner-up underlined. In the undefended case, all methods score highly because the task reduces to identifying a single site from a clean trace. The differences emerge once obfuscation is enabled. Under Walkie-Talkie, our model stays highly accurate (98.72 standing out against burst-moulding. Under padding-based defences, the gap narrows: RF [24] is slightly higher under WTF-PAD (96.58% vs. 95.84%) and Front (93.34% vs. 92.93%), but our approach remains close while performing consistently across settings. TrafficSliver is especially disruptive, yet our method achieves the best accuracy (84.05%). RegularTor, Tamaraw, and CS-BuFLO reduce accuracy for all attacks, with Tamaraw/CS-BuFLO also imposing substantial overhead. Averaged across defences (Avg Defended), our approach ranks first overall, followed by RF [24] and Tik-Tok [14].

C. Multitab

1) Closed World: We evaluate our method in a closed-world multitab setting to measure performance under concurrent browsing. Table III reports AUC, $P@K$, and $\mathrm{MAP@}K$ for 2–5 tabs, where we set K equal to the number of tabs for each trace (e.g., $K{=}4$ for a 4-tab trace) to keep comparisons consistent. As concurrency increases, all methods degrade, but our approach consistently delivers the strongest precision. In the 2-tab case, we achieve an AUC of 98.80 with $P@2{=}85.05$ and $\mathrm{MAP@}2{=}90.45$, improving over ARES [32] by 2.39 and 1.83, and over TMWF [26] by 11.59 in $\mathrm{MAP@}2$. This advantage persists as tabs grow: for 3 tabs we reach $P@3{=}79.21$ and $\mathrm{MAP@}3{=}87.46$ (both higher than ARES), and for 5 tabs we retain $P@5{=}76.64$ and $\mathrm{MAP@}5{=}85.35$, exceeding ARES by 1.98 and 2.77 with only a modest drop from 2 tabs. ARES is the closest competitor and attains slightly higher AUC, but its lower $P@K$ and $\mathrm{MAP@}K$ indicate less concentration of correct sites at the top of the ranking. Since multitab WF success hinges on recovering the true sites within the top few guesses, $P@K$ and $\mathrm{MAP@}K$ are the more meaningful criteria, which our method performs best across all tab settings.

2) Open World: Table IV presents the open-world evaluation of our method in the multitab setting. The results in Table IV closely mirror those in Table III, demonstrating our models' superior performance over SOTA methods. Across all tab settings, our model consistently achieves the strongest precision and ranking performance. In the 2-tab setting, we record a $P@2$ of 83.32 and a $\mathrm{MAP@}2$ of 88.94; ARES gets a $P@2$ of 82.44 and a $\mathrm{MAP@}2$ of 88.15, so we outperform it by 0.88 and 0.79. As the number of open tabs increases, which typically adds interference and degrades accuracy, our

TABLE III

COMPARISON OF EXISTING METHODS IN THE CLOSED-WORLD SETTING USING MULTITAB EVALUATION ON AUC, $P@K$, AND MAP@K ACROSS TAB COUNTS. THE MOST EFFECTIVE ATTACK BASED ON EACH METRIC IS SHOWN IN **BOLD**.

	2-tab			3-tab			4-tab			5-tab		
Attacks ↓	**AUC**	$P@2$	MAP@2	**AUC**	$P@3$	MAP@3	**AUC**	$P@4$	MAP@4	**AUC**	$P@5$	MAP@5
NetCLR	84.91	34.55	41.57	74.01	21.10	26.93	69.84	18.70	23.50	65.58	16.11	19.66
BAPM	93.52	52.86	62.22	87.20	38.44	49.24	83.90	35.44	45.24	80.04	30.77	39.31
DF	94.45	60.15	71.23	86.44	42.44	56.68	83.09	37.47	51.05	77.62	30.80	43.43
Tik-Tok	95.85	64.76	75.47	87.66	45.17	59.75	83.92	38.70	52.93	78.12	31.13	43.85
Var-CNN	96.12	65.52	75.29	90.61	51.23	65.46	88.16	47.15	60.56	84.24	39.44	52.26
TMWF	97.26	72.26	78.86	94.65	63.54	72.00	93.11	60.71	68.54	89.33	50.00	58.63
RF	94.81	64.11	72.94	88.05	47.10	59.10	85.20	43.43	56.11	79.85	33.83	44.67
ARES	**98.99**	82.66	88.62	**98.14**	78.47	86.10	**97.85**	**79.38**	**86.24**	**96.91**	74.66	82.58
Ours	98.80	**85.05**	**90.45**	97.77	**79.21**	**87.46**	97.11	77.32	**86.24**	96.64	**76.64**	**85.35**

TABLE IV

COMPARISON OF EXISTING METHODS IN THE OPEN-WORLD SETTING USING MULTITAB EVALUATION ON AUC, $P@K$, AND MAP@K ACROSS TAB COUNTS. THE MOST EFFECTIVE ATTACK BASED ON EACH METRIC IS SHOWN IN **BOLD**.

	2-tab			3-tab			4-tab			5-tab		
Attacks ↓	**AUC**	$P@2$	MAP@2	**AUC**	$P@3$	MAP@3	**AUC**	$P@4$	MAP@4	**AUC**	$P@5$	MAP@5
NetCLR	94.00	57.87	67.09	87.04	42.05	53.92	84.03	37.18	48.63	79.68	32.96	43.41
BAPM	93.31	52.02	61.64	87.01	37.99	48.17	83.64	35.03	44.98	79.83	31.03	39.51
DF	94.18	57.84	68.82	85.67	41.62	56.07	82.68	37.20	50.89	78.07	31.54	44.45
Tik-Tok	95.46	63.37	73.82	87.23	44.92	59.92	83.67	38.45	52.73	78.17	31.45	44.26
Var-CNN	97.24	70.27	79.62	92.24	57.20	70.49	87.33	45.39	58.17	75.25	26.31	35.74
TMWF	96.80	70.14	76.88	93.84	60.51	69.17	92.07	57.70	65.92	90.29	53.18	62.18
RF	94.24	62.80	71.50	88.09	47.40	60.31	85.77	43.21	55.91	81.85	37.19	48.35
ARES	**98.95**	82.44	88.15	**97.94**	77.79	85.61	**97.90**	79.03	85.89	**96.97**	74.86	83.18
Ours	98.68	**83.32**	**88.94**	97.82	**80.27**	**88.03**	97.21	**79.16**	**87.38**	96.81	**77.34**	**85.95**

TABLE V

GENERALIZATION TEST UNDER TAB MISMATCH: MODELS TRAINED ON n-TAB TRACES AND EVALUATED ON DIFFERENT TAB COUNTS. BOLD VALUES INDICATE THE BEST AUC PERFORMANCE IN EACH COLUMN.

	2-tab (Train)			3-tab (Train)			4-tab (Train)			5-tab (Train)		
Attacks ↓	3-tab	4-tab	5-tab	2-tab	4-tab	5-tab	2-tab	3-tab	5-tab	2-tab	3-tab	4-tab
NetCLR	68.71	63.05	60.18	74.90	62.88	59.60	69.85	65.27	58.88	66.55	63.01	59.57
BAPM	74.77	67.01	63.18	82.50	69.45	64.35	74.45	72.26	64.03	68.81	68.55	66.40
DF	75.55	67.55	63.87	85.01	69.40	64.55	80.75	74.09	64.18	79.31	70.88	67.21
Tik-Tok	76.59	68.82	64.67	86.27	70.67	65.85	81.55	70.85	64.65	79.84	72.31	67.72
Var-CNN	79.36	71.67	66.44	89.11	74.00	67.85	84.40	77.77	66.25	83.51	76.19	71.29
TMWF	75.80	67.21	63.05	83.67	71.32	65.75	75.91	72.85	65.02	72.86	68.85	66.65
RF	75.52	67.64	62.75	85.00	70.51	63.87	80.25	74.35	63.65	79.00	71.51	67.16
ARES	86.50	77.70	71.05	92.30	81.74	74.00	89.10	85.35	76.15	87.34	81.71	79.00
Ours	**87.89**	**79.80**	**73.08**	**93.81**	**84.87**	**77.23**	**90.97**	**87.82**	**78.64**	**89.01**	**84.77**	**81.94**

approach maintains its advantage. In the 3-tab setting, our model achieves a $P@3$ of 80.27, surpassing ARES by 2.48. We also attain a MAP@3 of 88.03, surpassing ARES by 2.42. In the 4-tab setting, we attain a $P@4$ of 79.16 and MAP@4 of 87.38, surpassing ARES by 0.13 and 1.49. Even under the challenging 5-tab setting, our method still leads with a $P@5$ of 77.34 and MAP@5 of 85.95, consistently outperforming ARES by noticeable margins (2.48 and 2.77), indicating strong robustness in the multitab setting.

3) Generalization of multitab Attack: Table V studies tab-mismatch generalisation by training each attack on a fixed tab count (2–5) and testing on the others, reflecting the realistic case where the number of active tabs at inference time is unknown. We report AUC because it is threshold-free and comparable across tab counts, and we use a warm-start protocol: we train once per tab count, select the best checkpoint, and evaluate its weights on other tab settings without fine-tuning, while keeping the architecture and hyperparameters fixed. As expected, mismatch degrades all methods, and the drop grows with the train–test gap (often 10–15 AUC points from 2 to 5 tabs), but our method is consistently the most stable. Across all training regimes (2–5 tabs), it achieves the highest AUC on every mismatched test set, outperforming the strongest baseline ARES, and shows the smallest losses even under severe mismatch (e.g., 2→5 and 5→2).

D. Explainability

This section provides insight into what drives our attack by applying post-hoc Integrated Gradients (IG) [11] to a

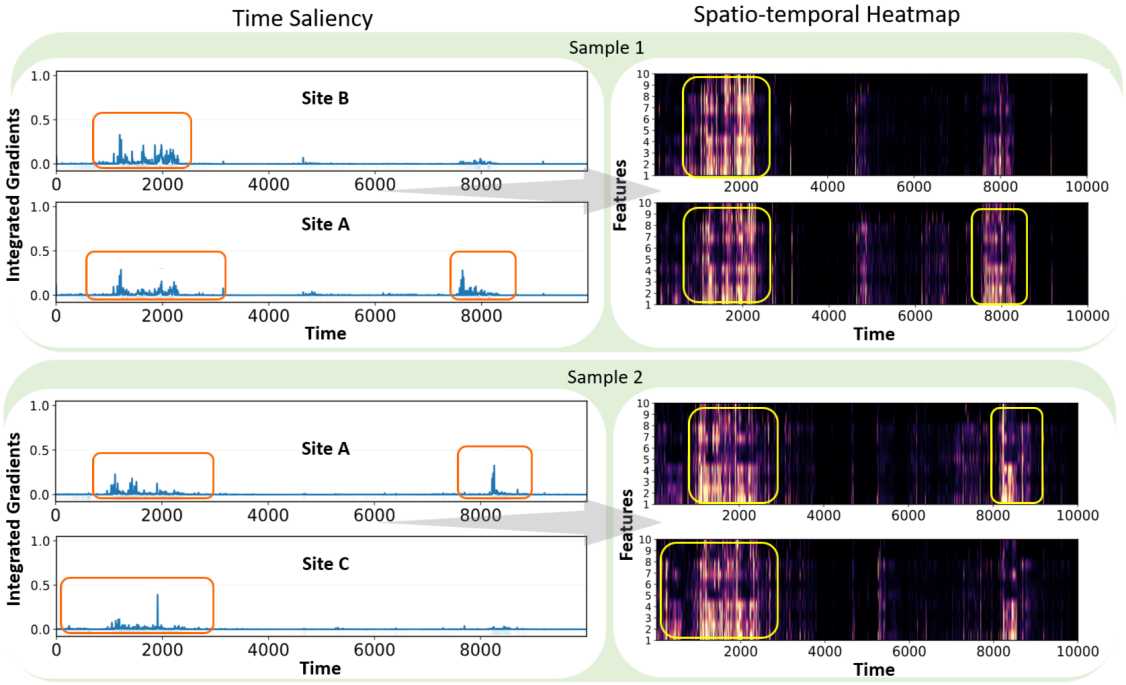

Fig. 3. Time saliency (left) and spatio-temporal heatmaps (right) for two 2-tab traces. Class A shows consistent salient intervals across samples, while classes B and C emphasise distinct regions, illustrating intra-class consistency and inter-class separation.

trained model, revealing which parts of a trace most influence its predictions. IG attributes the change in a target-class logit $f_k(x)$ to individual input elements by integrating input gradients along the straight-line path from a baseline x' (we use an all-zero sequence to represent absence of signal) to the observed input $x \in \mathbb{R}^{C\times L}$. The attribution for channel c at time t is

$$\mathrm{IG}_{c,t}(x) = \left(x_{c,t} - x'_{c,t}\right) \int_0^1 \frac{\partial f_k(x' + \alpha(x - x'))}{\partial x_{c,t}}\, d\alpha. \quad (3)$$

IG satisfies completeness: summing $\mathrm{IG}_{c,t}(x)$ over all c, t recovers $f_k(x) - f_k(x')$ up to numerical error, yielding a faithful decomposition of the model's decision into per-element contributions.

1) Time Saliency: In a passive setting, an attacker may want to know *when* the model's evidence for class k is concentrated. We capture this with *time saliency*, $S_t = \sum_{c=1}^{C} |\mathrm{IG}_{c,t}|$, which aggregates absolute IG across channels at time t. For visualisation, we compute S_t per trace, average it over the test set, and smooth with a moving average. In the closed-world split of two tabs, Figure 4 shows that the model is most based on early time-steps, especially 0–4000. We validate this trend by training on only the early (0–4000) or late (4000–10000) segment. Table VI shows that MAP@2 drops from 83.65 to 62.19 when the early segment is removed, confirming that early time steps carry most of the discriminative signal, while later segments retain some information for a subset of classes.

2) Spatio-Temporal Heatmap: We visualise feature importance over time using spatio-temporal IG heatmaps. The

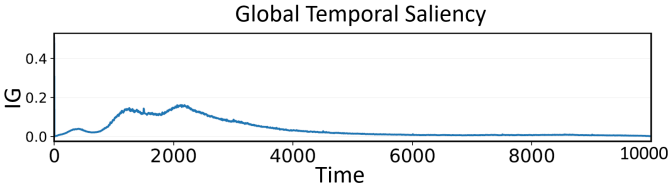

Fig. 4. Global temporal saliency indicating the classifier relies heavily on early trace segments.

TABLE VI
MAP@2 WHEN TRAINING ON DIFFERENT VISIBLE TRACE SEGMENTS.

Visible trace segment	MAP@2
0–4000	**83.65**
4000–10000	62.19

attribution matrix $\mathrm{IG} \in \mathbb{R}^{C\times T}$ assigns a contribution to each feature c at time t for class k; high-intensity regions indicate where specific features strongly drive the prediction. Combined with time saliency, these heatmaps provide trace-level explanations.

Intra-class consistency and Inter-class separation. Figure 3 illustrates both intra-class consistency and inter-class separation using two randomly selected 2-tab traces that include the target site (Site A): sample 1 contains Sites A and B (A ranked 2) and sample 2 contains Sites A and C (A ranked 1). For Site A, the time-saliency curves peak in similar regions across both traces (around 1000–2500 and near 8000), and the heatmaps show a stable feature signature (stronger activation

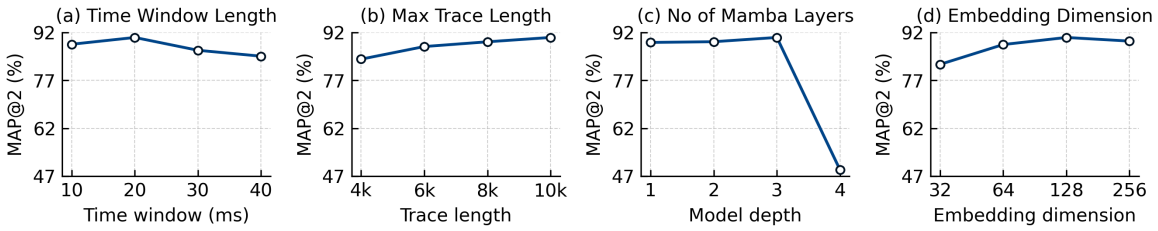

Fig. 5. Impact of key hyperparameters on closed-world performance (MAP@2). We vary (a) time window length, (b) maximum trace length, (c) number of Mamba layers (model depth), and (d) embedding dimension, while keeping all other settings fixed.

TABLE VII
ABLATION STUDIES

Model		Packets	Burst			Time	MAP@2
HTM	GM	Counts	Counts	Medians	Max/Min	IAT	
✓	✓	✓	✓	✓	✓	✓	**90.45**
✓	✗	✓	✓	✓	✓	✓	85.49
✗	✓	✓	✓	✓	✓	✓	84.87
✓	✓	✓	✗	✗	✗	✗	87.50
✓	✓	✗	✓	✗	✗	✗	84.30
✓	✓	✗	✗	✓	✗	✗	85.22
✓	✓	✗	✗	✗	✓	✗	81.60
✓	✓	✗	✓	✓	✓	✗	87.32
✓	✓	✗	✗	✗	✗	✓	82.75

in features 1–4 and 7–8 than 5–6), indicating consistent trace-level evidence and helping explain why later segments can still contribute (Table VI). At the same time, co-occurring sites exhibit distinct evidence: in sample 1, Site B concentrates almost entirely in 1000–2000 and shows little late activity, while Site A retains a clear late peak and broader feature activation; in sample 2, Site C is similarly dominated by early saliency with minimal late signal. These differences demonstrate that the model assigns label-specific temporal and feature-level patterns within the same multitab trace rather than conflating sites into a single mixed representation.

3) Ablation studies: In our ablation study, we quantify the contribution of major architectural modules and feature groups by training variants on the 2-tab closed-world split and reporting MAP@2 in Table VII. The full model performs best (90.45), and removing either modelling component reduces accuracy: using only hierarchical temporal modelling (HTM) drops to 85.49, while using only the global modelling (GM) module drops further, indicating that global context alone is insufficient and that HTM provides the stronger foundation. Feature ablations show that packet-count features are the most informative single group (87.50), with burst-level features nearly matching (87.32), implying that leakage persists both at packet level and in burst structure. Within burst descriptors, median burst size is most discriminative (85.22), followed by burst counts (84.30), while max/min burst size is weaker (81.60). Inter-arrival times alone yield the lowest score (82.75), suggesting timing remains useful but is less informative than packet and burst statistics. Overall, both modules and features contribute, with HTM dominating the modelling stack and packet/burst statistics dominating the feature space.

V. DISCUSSION

Parameter Sensitivity. Figure 5 shows the impact of key hyperparameters on the 2-tab closed-world split, varying one factor at a time while keeping others fixed. In Fig. 5a, MAP@2 peaks at a 20 ms segmentation window and declines for larger windows, suggesting that coarse windows smooth away useful WF structure. Fig. 5b shows that increasing the maximum trace length from 4k to 10k packets steadily improves performance, indicating the model benefits from additional context in multitab traces. Fig. 5c varies Mamba depth: performance improves up to three layers but drops sharply at four, implying moderate depth is sufficient. Fig. 5d shows gains as the embedding dimension increases to 128, followed by a small decline at 256, making 128 a good capacity–regularization trade-off. Overall, aside from depth, performance varies only mildly across the explored ranges, indicating limited sensitivity to these hyperparameters.

MAP@2 vs. Trace Length (Ours vs. ARES). In multitab

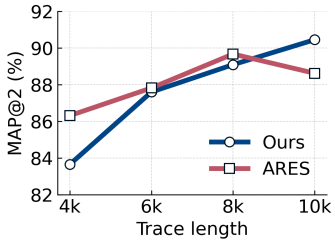

Fig. 6. MAP@2 comparison on trace length for our model and ARES

WF, models should exploit additional observations as traces grow [1]. Figure 6 plots MAP@2 versus trace length for our model and ARES [32]. ARES is competitive on short traces and we match it at medium lengths, but once the trace exceeds ∼8000 packets its performance degrades, indicating reduced discriminative power as more context is added. In contrast, our model continues to improve with longer traces and achieves higher MAP@2. This behaviour is consistent with the difficulty of scaling transformer backbones to long sequences, where quadratic attention and representation drift can hurt performance, suggesting caution when applying transformer-based WF models to long traces.

Countermeasure. Our attack leverages statistical structure in packet directions, bursts, and timing, so an effective

defence should reduce cross-site variation especially highly site-specific bursts that disproportionately aid classification. Rather than targeting a specific model, a defence could use a lightweight proxy classifier (e.g., a compact CNN) to flag unusually distinctive bursts and then apply targeted padding, burst reshaping, and controlled timing perturbations to map traffic onto a small set of generic templates that balance overhead while suppressing both fine and coarse-grained signals, even when attackers train on defended traces. This has been scheduled for our future research.

VI. Conclusion

We present a novel WF attack designed for realistic multitab browsing. We build on limitations of current methods that simplify the WF task and design a classifier that outperforms state-of-the-art in mutiple settings. Our method combines a dilated CNN, which captures patterns in local trace segments, with a Mamba stack that aggregates information over the whole sequence. Experiments show that our method consistently outperforms SOTA in both precision and generalization, and remains effective as the number of tabs and the amount of training data grow, making it well suited to iterative deployment. Finally, our temporal saliency and spatio-temporal heatmap analysis provide interpretable evidence on when and where the model finds discriminative signals, clarifying how our WF attacks operate.

References

[1] Deng, X., Zhao, R., Wang, Y., Zhan, M., Xue, Z. & Wang, Y. Countmamba: A Generalized Website Fingerprinting Attack via Coarse-Grained Representation and Fine-Grained Prediction. *2025 IEEE Symposium On Security And Privacy (SP)*. pp. 1419-1437 (2025)

[2] X. Cai, R. Nithyanand, T. Wang, R. Johnson, and I. Goldberg, "A systematic approach to developing and evaluating website fingerprinting defenses," *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pp. 227–238, 2014.

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

[4] J. Hayes, and G. Danezis, "k-fingerprinting: A robust scalable website fingerprinting technique," *25th USENIX Security Symposium (USENIX Security 16)*, pp. 1187–1203, 2016.

[5] A. Panchenko, F. Lanze, J. Pennekamp, T. Engel, A. Zinnen, M. Henze, and K. Wehrle, "Website Fingerprinting at Internet Scale.," *NDSS*, vol. 1, pp. 23477, 2016.

[6] M. Juarez, M. Imani, M. Perry, C. Diaz, and M. Wright, "Toward an efficient website fingerprinting defense," *European Symposium on Research in Computer Security*, pp. 27–46, 2016.

[7] T. Wang, and I. Goldberg, {Walkie-Talkie}: An efficient defense against passive website fingerprinting attacks, pp. 1375–1390, 2017.

[8] J. Yan, and J. Kaur, "Feature selection for website fingerprinting," *Proceedings on Privacy Enhancing Technologies*, 2018.

[9] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep fingerprinting: Undermining website fingerprinting defenses with deep learning," *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pp. 1928–1943, 2018.

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

[11] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," *International conference on machine learning*, pp. 3319–3328, 2017.

[12] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: A data-efficient website fingerprinting attack based on deep learning," *arXiv preprint arXiv:1802.10215*, 2018.

[13] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning," *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1131–1148, 2019.

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

[15] W. De la Cadena, A. Mitseva, J. Hiller, J. Pennekamp, S. Reuter, J. Filter, T. Engel, K. Wehrle, and A. Panchenko, "Trafficsliver: Fighting website fingerprinting attacks with traffic splitting," *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1971–1985, 2020.

[16] J. Gong, and T. Wang, "Zero-delay lightweight defenses against website fingerprinting," *29th USENIX Security Symposium (USENIX Security 20)*, pp. 717–734, 2020.

[17] J. K. Holland, and N. Hopper, "Regulator: A straightforward website fingerprinting defense," *arXiv preprint arXiv:2012.06609*, 2020.

[18] M. S. Rahman, M. Imani, N. Mathews, and M. Wright, "Mockingbird: Defending against deep-learning-based website fingerprinting attacks with adversarial traces," *IEEE TIFS*, vol. 16, pp. 1594–1609, 2020.IEEE,

[19] M. Nasr, A. Bahramali, and A. Houmansadr, Defeating {DNN-Based} traffic analysis systems in {Real-Time} with blind adversarial perturbations, *30th USENIX Security Symposium (USENIX Security 21)*, pp. 2705–2722, 2021.

[20] S. E. Oh, N. Mathews, M. S. Rahman, M. Wright, and N. Hopper, "GANDaLF: GAN for data-limited fingerprinting," *Proceedings on privacy enhancing technologies*, vol. 2021, no. 2, 2021.

[21] J. Gong, W. Zhang, C. Zhang, and T. Wang, "Surakav: Generating realistic traces for a strong website fingerprinting defense," *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 1558–1573, 2022.

[22] Tor, "The-Tor-Project," *https://metrics.torproject.org/news.html*, 2023.

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

[24] M. Shen, K. Ji, Z. Gao, Q. Li, L. Zhu, and K. Xu, "Subverting website fingerprinting defenses with robust traffic representation," *32nd USENIX Security Symposium (USENIX Security 23)*, pp. 607–624, 2023.

[25] X. Deng, Q. Yin, Z. Liu, X. Zhao, Q. Li, M. Xu, K. Xu, and J. Wu, "Robust multi-tab website fingerprinting attacks in the wild," *2023 IEEE symposium on security and privacy (SP)*, pp. 1005–1022, 2023.

[26] Z. Jin, T. Lu, S. Luo, and J. Shang, "Transformer-based model for multi-tab website fingerprinting attack," *2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1050–1064, 2023.

[27] X. Deng, Q. Li, and K. Xu, "Robust and reliable early-stage website fingerprinting attacks via spatial-temporal distribution analysis," *ACM CCS 2024*, pp. 1997–2011, 2024.

[28] X. Zhao, X. Deng, Q. Li, Y. Liu, Z. Liu, K. Sun, and K. Xu, "Towards fine-grained webpage fingerprinting at scale," *ACM CCS 2024*, pp. 423–436, 2024.

[29] N. Mathews, J. K. Holland, N. Hopper, and M. Wright, "Laserbeak: Evolving website fingerprinting attacks with attention and multi-channel feature representation," *IEEE TIFS*, 2024.IEEE,

[30] A. Mitseva, and A. Panchenko, "Stop, don't click here anymore: boosting website fingerprinting by considering sets of subpages," *33rd USENIX Security Symposium (USENIX Security 24)*, pp. 4139–4156, 2024.

[31] Z. Guan, G. Xiong, G. Gou, Z. Li, M. Cui, and C. Liu, "BAPM: block attention profiling model for multi-tab website fingerprinting attacks on tor," *Proceedings of the 37th Annual Computer Security Applications Conference*, pp. 248–259, 2021.

[32] X. Deng, X. Zhao, Q. Yin, Z. Liu, Q. Li, M. Xu, K. Xu, and J. Wu, "Towards Robust Multi-tab Website Fingerprinting," *arXiv preprint arXiv:2501.12622*, 2025.

[33] X. Cai, R. Nithyanand, and R. Johnson, "Cs-buflo: A congestion sensitive website fingerprinting defense," *Proceedings of the 13th Workshop on Privacy in the Electronic Society*, pp. 121–130, 2014.

[34] W. Meng, C. Ma, M. Ding, C. Ge, Y. Qian, and T. Xiang, "Beyond single tabs: A transformative few-shot approach to multi-tab website fingerprinting attacks," *Web Conference 2025*, pp. 1068–1077, 2025.

[35] Y. Xie, J. Feng, W. Huang, Y. Zhang, X. Sun, X. Chen, and X. Luo, "Contrastive fingerprinting: A novel website fingerprinting attack over few-shot traces," *Proceedings of the ACM Web Conference 2024*, pp. 1203–1214, 2024.