Evaluating Boundary Restriction Methods Against Hardware Transient Faults on Website Fingerprinting Attacks

Chaoyue Ren, Yixu Yu, Ran Wen, *Jiajia Jiao
College of Information Engineering
Shanghai Maritime University, Shanghai, China
{202330310244, 202430310285, 202230310052}@stu.shmtu.edu.cn, jiaojiajia@shmtu.edu.cn

***Abstract*—Boundary restriction methods are increasingly used to mitigate the impacts of transient hardware faults due to their low overhead and ease of use. Besides the typical safety-critical applications such as self-driving and health-care, the machine learning-assisted website fingerprinting (WF) attacks, which aim to infer sensitive user activities from encrypted traffic patterns, require high computational reliability. However, transient hardware faults (e.g., bit-flips) may degrade attack accuracy by distorting feature extraction. Therefore, this paper aims to enhance the robustness of website fingerprinting (WF) attack models against hardware transient faults by evaluating boundary restriction methods. Three boundary restriction methods including layer-level clipping, neuron-level smooth restrictions, and hybrid strategies are selected for mitigating fault propagation in the WF learning models. The evaluation of fault containment efficacy is through hardware-calibrated fault injection across convolutional (Conv), activation, and fully-connected (FC) layers. The experiments on interrupt-driven datasets reveal that layer-level approaches maintain inference accuracy yet show limited fault resilience. Neuron-level methods reduce fault propagation but degrade feature extraction, with accuracy drops of 2.78% in fault-free scenarios and nonlinear performance decay under increasing Bit Error Rates (BERs). Hybrid strategies balance these tradeoffs by selectively integrating layer-level and neuron-level restrictions, achieving 35.3% accuracy under BER equal to 3×10^{-6} while maintaining moderate fault-free accuracy (0.32% drop). The comprehensive results analysis provides design principles for integrating boundary restrictions into reliable WF attacks while harmonizing hardware reliability and algorithmic precision.**

***Index Terms*—Boundary Restriction Methods, Hardware Transient Faults, Website Fingerprinting Attacks, Fault Propagation Mitigation**

I. Introduction

Deep Neural Networks (DNNs) have revolutionized applications ranging from computer vision to natural language processing, supporting mission-critical systems such as autonomous driving and healthcare diagnostics, as well as privacy-sensitive tasks like website fingerprinting (WF) [1], [2]. It is important to note that the impact of such faults in website fingerprinting differs significantly from other DNN applications. While tasks like image classification benefit from inherent redundancy and spatial invariance that can mask minor perturbations, website fingerprinting relies on subtle and time-sensitive traffic features. Even slight distortions can lead to false positives or missed detections, thereby critically affecting the model's performance. However, ensuring the reliability of DNNs against hardware transient faults, such as random bit-flips caused by cosmic radiation or voltage instability, has become a pressing concern. In the context of website fingerprinting, these transient faults can distort the feature extraction process—particularly in spatial layers—leading to significant misclassification of traffic patterns and ultimately reducing the accuracy of inferring visited websites [3].

Although existing DNNs fault-tolerant designs (e.g., retraining with fault injection [4], modular redundancy [5]) mitigate transient faults, boundary restriction methods have emerged as a lightweight alternative. These methods constrain the activation values within predefined ranges, thereby limiting error propagation [6]. However, their adaptation to privacy-sensitive tasks continues to pose significant challenges: 1) overly restrictive bounds may suppress discriminative features, 2) loose thresholds may permit fault accumulation. Therefore, the tradeoff becomes particularly critical when models are required to simultaneously preserve adversarial robustness and tolerate hardware distortions, especially for WF, the focus of this paper.

WF is a representative task where models combining Conv layers (for local traffic feature extraction) with LSTM networks (for temporal dependency modeling) must contend with both adversarial distortions and transient hardware faults. WF attack, inferring visited websites from metadata (e.g., packet timing/size sequences) [7], epitomizes this conflict.

To quantify real-world impacts on WF attacks, we analyze models trained on interrupt-driven data. Interrupt-driven data is a key target for WF attacks, given the strict real-time constraints of interrupt-driven systems, which are paradoxically coupled with metadata vulnerabilities. In this context, hardware faults introduce dual risks: 1) distorted features may cause false privacy breaches (misclassifying benign interrupt patterns as sensitive) or 2) missed attacks (failing to detect vulnerabilities).

* Corresponding author: Jiajia Jiao

Modern WF models rely on sequential architectures such as LSTMs [8] and temporal CNNs [19] to capture subtle traffic patterns. In interrupt-driven signal analysis, the prevalent hybrid CNN-LSTM architecture [20], [21] combines CNNs to capture local interrupt sequence correlations with LSTMs that model system-level temporal dependencies, enabling robust feature extraction even in the presence of operational noise. However, these architectures introduce a critical vulnerability: the cascading nature of temporal faults in LSTMs amplifies hardware-induced errors, while inherent security measures, such as interrupt timing randomization mandated by hardware isolation [9], intentionally obscure input patterns. This coupling of stochastic hardware noise and security-induced signal distortions creates an unresolved tension in boundary restriction design. Critically, while WF attacks aim to bypass privacy defenses, their computational reliability must also be preserved. Strict containment thresholds risk suppressing time-sensitive discriminative features critical for interrupt pattern recognition, whereas lenient bounds permit progressive error accumulation.

Existing boundary restrictions, predominantly optimized for activation layers in vision tasks (e.g., ReLU clipping in Convolutional networks [6]), fail to address these dynamics. Unlike spatially localized image features, LSTM rely on temporal gate mechanisms (input, forget, and output gates) whose state transitions govern sequential feature extraction. This process fundamentally misaligned with static activation value clipping. Compounding this limitation, fault injection tools like PyTorchFI [10] lack granular control over LSTM internals (e.g., cell state corruption), leaving WF-specific error propagation mechanisms in boundary-restricted models poorly understood. To bridge these gaps (the lack of fault injection tools for hybrid CNN-LSTM architectures and domain-agnostic boundary restrictions in metadata-driven tasks), this paper undertakes the first comprehensive evaluation of boundary restriction methods in LSTM-enhanced WF models under transient faults, quantifying their impacts on both fault tolerance and adversarial robustness.

Our main contributions are as follows:

- Extended Fault Injection Framework. Augmenting PyTorchFI with hierarchical fault injection capabilities for hybrid CNN-LSTM architectures, which enable bit-flip emulation in LSTM gates (input/forget/output) and CNN filter weights, critical components for interrupt analysis.
- Impact of Layer Types on Fault Tolerance. The experiments on interrupt-driven datasets reveal significant differences in fault sensitivity across layer types. Conv and FC layers exhibit severe accuracy drops of 50.82% and 47.99%, respectively, under transient faults. In contrast, LSTM layers show remarkable resilience, with only a minimal 0.07% accuracy reduction, highlighting their intrinsic fault tolerance and identifying spatial layers (Conv and FC) as critical reliability bottlenecks in hybrid CNN-LSTM architectures.
- Boundary Restriction Benchmark. Systematically evaluating layer-level, neuron-level, and hybrid boundary restriction methods. Layer-level restrictions degrades accuracy to 30.9% under BER equal to 3×10^{-6}. In contrast, neuron-level restrictions improve accuracy by 5.18% relative to layer-level restrictions at the same BER but degrade fault-free accuracy by 2.78%. Hybrid strategies like ProAct achieve a balanced trade-off, maintaining moderate fault-free accuracy (0.32% drop) and controlled fault degradation, achieving 35.3% accuracy under BER equal to 3×10^{-6}. These findings emphasize the need for selective boundary restrictions to balance fault resilience and feature preservation.
- Practical Guidelines. Characterizing the tradeoff between fault tolerance and feature preservation across layers reveals that selectively applying boundary restrictions to spatial layers while preserving LSTM gate dynamics offers the optimal balance, which is a critical insight for deploying reliable WF systems. Future work can extend these findings by exploring lightweight restrictions on LSTM layers to further enhance robustness without significantly compromising accuracy.

II. Related work

A. Limitations of Boundary Restriction Methods in Website Fingerprinting

Boundary restriction methods, designed to suppress fault propagation by constraining activation values, exhibit critical limitations when applied to WF attacks.

Static approaches like Ranger [11], which rely on layer-level thresholds derived from empirical training data, fail to adapt to the dynamic temporal patterns inherent in encrypted traffic analysis. For instance, fixed clipping bounds optimized for Conv layers in vision tasks misalign with the sequential dependencies in LSTM-based WF models, where transient hardware faults in gate (e.g., input/forget gates) can cascade through temporal states.

Dynamic methods such as FTClip [12] and FitAct [13] attempt to optimize thresholds via gradient search but neglect neuron-level vulnerabilities in LSTM cells, which is vital for capturing long-term traffic correlations.

Hybrid strategies like ProAct [14], which combine layer-level and neuron-level constraints, assume unidirectional error propagation and thus falter in bidirectional architectures common to WF models (e.g., CNN-LSTM hybrids). Crucially, existing methods focus solely on activation layers (e.g., ReLU), leaving non-activation components (e.g., LSTM cell state updates) unregulated. This oversight allows hardware-induced errors in temporal operations to propagate unchecked, undermining both fault tolerance and WF accuracy.

B. Reliability Assessment in Privacy-Sensitive Tasks

Evaluating boundary restriction methods for WF attacks faces two interrelated obstacles: 1) lack of temporal fault assessment frameworks and 2) data scarcity under privacy constraints.

Traditional fault injection frameworks like PyTorchFI [10] and TensorFI [15], designed for CNNs and fully connected layers, lack support for sequential modules such as LSTM gates, critical for modeling interrupt timing dependencies. This tooling gap leaves temporal fault propagation paths unexamined, particularly errors in LSTM input/forget gates that distort long-term interrupt correlations. Furthermore, reliability metrics in prior work (e.g., classification accuracy) fail to capture operational risks in interrupt monitoring, such as false positives in critical interrupt detection or missed events due to fault-induced feature corruption.

Compounding these methodological gaps is the absence of open, standardized WF datasets. Unlike image recognition domains benefiting from large, diverse benchmarks (e.g., ImageNet), WF research faces inherent privacy and data collection challenges. Real-world network traces containing sensitive user browsing behaviors are rarely publicly available due to ethical constraints, forcing researchers to manually collect interrupt timing traces from target websites, which is prone to scalability and consistency issues. Acquiring website-specific timing data requires overcoming privacy-preserving countermeasures (e.g., encrypted traffic, randomized packet padding) that obscure interrupt patterns. Second, the absence of standardized collection protocols (e.g., uniform hardware environments, browser configurations) leads to dataset heterogeneity, making cross-study comparisons unreliable. For example, interrupt traces collected in controlled lab settings often ignore deployment variables, such as cross-device timing jitter or background network noise, and these distortions significantly degrade clipping threshold calibration. Consequently, models trained on limited self-collected datasets tend to overfit to localized noise patterns, failing to generalize under dynamic network conditions. This data scarcity fundamentally limits the reliability of boundary restrictions in balancing false acceptance (admitting adversarial noise) and false rejection (discarding discriminative features).

III. Experiments configuration

A. The Details of Model and Dataset

Model Structure. Following the LSTM framework described in [17], the classifier integrates temporal and spatial modules. A 32-unit LSTM layer captures interrupt sequence dependencies, followed by two 1D convolutional blocks (256 filters, stride=3, each followed by ReLU activation) to extract localized timing deviations. Spatial features are processed through max-pooling layers (pool size=4) and dropout regularization (rate=0.7), before passing to the final softmax classification layer. The model is trained using the Adam optimizer (learning rate=0.001) and evaluated via 10-fold cross-validation. To evaluate the resilience of WF attack models under hardware faults, all boundary restriction methods (Ranger, FitAct, FTClip, ProAct) are integrated into the CNN-LSTM architecture. These methods suppress error propagation in spatial layers (Conv/FC) while preserving the temporal dynamics of LSTM, ensuring that attack accuracy is maintained even under hardware-induced weight corruptions.

Interrupt-Driven Dataset. The study introduces an interrupt-driven dataset that captures hardware-level timing metadata to evaluate the robustness of website fingerprinting models under transient hardware faults. The dataset comprises 10,000 traces from 100 monitored websites (e.g., e-commerce and streaming platforms), with each website containing 100 full page-load sessions. Each trace sequences interrupt timing micropatterns triggered by browser activities. To standardize variable interrupt event counts, raw temporal sequences are padded or truncated, preserving the correlation between hardware stalls and browser rendering states. To ensure rigorous evaluation, we partition the dataset into 90% for training and 10% for testing, maintaining class balance across partitions through stratified sampling based on website categories. This split protocol mitigates overfitting while preserving intra-website behavioral diversity in both subsets.

B. Fault propagation mechanism and Fault test framework

Assumptions of Hardware Transient Faults. In DNN accelerators, weight corruption caused by transient hardware faults (e.g., bit-flips in memory) critically impacts reliability due to its persistent propagation. These faults are modeled as random bit-flips in the exponent bits of floating-point weight representations, simulating severe perturbations (2^2–2^{64} × the original value) that amplify activation errors across layers. Unlike activation faults, which are often self-corrected by non-linear operations like ReLU or normalization layers, weight faults persistently distort computations throughout inference. For instance, a single bit-flip in a weight's exponent disrupts the scaling factor in floating-point arithmetic, significantly altering convolutional or matrix product outputs. These corrupted activations propagate spatially in early layers and accumulate locally in later layers, disproportionately affecting output logits. As shown in [18], the lack of inherent redundancy in weight parameters makes their faults irrecoverable, leading to cascading misclassifications even under minor errors. This aligns with our focus on weight corruption as the primary failure mode in safety-critical applications.

Fault Model in Experiments. This evaluation framework uses PyTorchFI [10] to simulate transient weight faults by injecting bit-flips into predefined exponent bits of floating-point parameters during inference phase. Fault severity is modulated through six BERs conditions (range from 10^{-7} to 3×10^{-5}), representing varying fault probabilities. To ensure statistical validity, each BER is tested over 50 trials with persistent weight corruption. The framework benchmarks Top-1 accuracy against fault-free baselines, quantifying resilience degradation.

Fault Model in LSTM. To broaden applicability, the extended PyTorchFI is leveraged to support fault injection in LSTM layers, critical for sequence modeling tasks. Specifically, the enhanced PyTorchFI is used to dynamically parse the internal structure of LSTM layers. We model hardware-induced transient bit-flips in LSTM layers by targeting two core weight matrices:

$$W_{ih} \in R^{H_{\text{in}} \times 4H_{\text{out}}} \tag{1}$$

$$W_{hh} \in R^{H_{\text{out}} \times 4H_{\text{out}}} \tag{2}$$

where $\boldsymbol{H_{in}}$ is the input dimension and $\boldsymbol{H_{out}}$ is the hidden state dimension. Here, $\boldsymbol{R}$ denotes real-valued matrices, and the term $\boldsymbol{4H_{out}}$ reflects the concatenated weights of the input, forget, cell, and output gates following PyTorch's LSTM implementation.

The number of bit-flips in each matrix is determined by a soft error rate $\boldsymbol{p_{sdc}}$, tensor dimensions, and hardware bit-width (e.g., 32-bit floating-point). For $\boldsymbol{W_{ih}}$, the fault count is:

$$N_{\text{fault}}^{(ih)} = [H_{\text{in}} \times 4H_{\text{out}} \times \text{bits} \times p_{\text{sdc}}] \tag{3}$$

Similar to $\boldsymbol{W_{hh}}$:

$$N_{\text{fault}}^{(hh)} = [H_{\text{in}} \times 4H_{\text{out}} \times \text{bits} \times p_{\text{sdc}}] \tag{4}$$

where bits represent the data bit-width.

During injection phase, we uniformly sample elements from $\boldsymbol{W_{ih}}$ and $\boldsymbol{W_{hh}}$ (defined in (1),(2)) to perturb their floating-point representations. This extension enables rigorous resilience testing for recurrent architectures by capturing fault propagation through hidden states and sequential dependencies. Fault injection targets both forward and recurrent computations, ensuring comprehensive coverage of LSTM vulnerabilities to bit-level perturbations.

C. SOTA Boundary Restriction Methods

In order to comprehensively evaluate the boundary restriction methods for LSTM-enhanced WF modele under hardware transient faults, four representative state-of-the-art boundary restriction methods are selected for comparative experiments.

Ranger [11] enforces static boundary restrictions to limit activation values by randomly sampling approximately 20% of the training data and calculating the maximum activation observed. This maximum value is used to determine the boundary, and the resulting boundaries are applied as hard truncation, zeroing out activations that exceed the limit. Ranger has two applied strategies, a global boundary is set based on the highest activation during validation in the Ranger layer version, and each neuron has its own boundary derived from past activation peaks, providing more fine-grained control in the Ranger neuron version.

FitAct [13] converts boundary restriction into learnable parameters. It combines ReLU with boundary control units to adjust activations according to neuron-specific boundary biases. Meanwhile, it calculates the maximum activation value for each neuron using 100% of the training data. To balance efficiency and accuracy, it adjusts the activation boundary values using 1%-10% of the original training data (depending on model size). This adjustment is made without changing the weight parameters, focusing only on modifying the activation function's parameters. The boundary values are optimized during backpropagation, enabling adaptive control of activations without relying on external validation or heuristic methods.

FTClip [12] introduces a dynamic boundary optimization method combined with fault simulation. It uses binary search to fine-tune the boundaries for each layer within a predetermined range, based on model accuracy under fault injection. Then, the boundaries with minimized accuracy loss are chosen to ensure robust activation limits. FTClip achieves fault-tolerance optimization for pre-trained models using a small validation set (e.g., a few hundred images), without the need for original training data or network retraining.

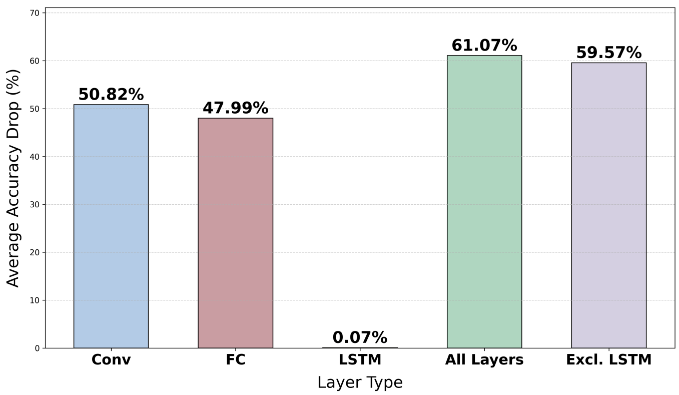


Fig. 1: Average Accuracy Drop by Layer Type.

ProAct [14] presents a progressive training framework that integrates both layer-level and neuron-level boundary restrictions. During the preprocessing stage, it samples about 5% of the validation data to initialize the truncation thresholds. Initially, layer-specific truncation is applied to early layers, optimizing boundaries through knowledge distillation. During training stage, ProAct shifts to neuron-level truncation with trainable boundaries, optimized through end-to-end training. This framework adapts boundaries throughout the training process, balancing computational cost and model reliability while integrating both hierarchical and neuron-level restrictions.

IV. Results and Analysis

A. Impact of Different Layer Types on CNN-LSTM Fault Sensitivity

Fault sensitivity is evaluated across various layer types in a hybrid CNN-LSTM model by injecting transient bit-flips under six bit-error rates (BERs). The baseline model (CNN-LSTM without fault injection) achieves 93.5% Top-1 accuracy on interrupt-driven sequence classification.

The average accuracy drop across different layer types. As shown in (5), the accuracy drop illustrates the accuracy loss between the baseline (without faults) and different layers (with faults), while the average accuracy drop represents the accuracy drop across six BERs conditions. As illustrated in Fig. 1, the fault-induced degradation varies dramatically across various layer types, exposing critical reliability bottlenecks in hybrid architectures. Conv layers and FC layers dominate accuracy reduction, with average accuracy drops of 50.82% and 47.99%, respectively. This significant degradation in the spatial layers critically impairs the extraction of discriminative traffic patterns that are essential for effective website fingerprinting. Although LSTM layers exhibit remarkable resilience with only a minimal average accuracy drop of 0.07%, errors originating in the spatial layers may still propagate and cumulatively affect the overall fingerprinting performance. This

TABLE I: Layer-Specific Accuracy Under Variable BERs

Layer	**BERs Condition**					
	$\mathbf{1 \times 10^{-7}}$	$\mathbf{3 \times 10^{-7}}$	$\mathbf{1 \times 10^{-6}}$	$\mathbf{3 \times 10^{-6}}$	$\mathbf{1 \times 10^{-5}}$	$\mathbf{3 \times 10^{-5}}$
Conv	91.6%	83.9%	65.6%	29.5%	4.23%	1.05%
FC	90.8%	86.8%	71.8%	36.3%	4.99%	1.11%
LSTM	93.5%	93.5%	93.5%	93.4%	93.4%	93.3%
All Layers	87.1%	78.0%	41.5%	9.76%	1.03%	1.00%
Excl. LSTM	87.1%	78.0%	49.0%	10.7%	1.02%	1.01%

Excl. LSTM: The result marked ”Excl. LSTM” means that only the non-LSTM layers (Conv and FC) are injected with weight errors, while keeping the LSTM layer weights undisturbed.
BERs Condition: Bit-Error Rates used in the experiment ranged from 1×10^{-7} (approximate random noise) to 3×10^{-5} (typical hardware transient faults) to simulate the transient fault scenario.

stark contrast highlights the catastrophic vulnerability of spatial layers compared to the intrinsic fault tolerance of LSTM layers. When faults are injected into all layers, the model's average accuracy drop decreases by 61.09%, underscoring the cascading propagation of errors across the architecture.

$$Accuracy_{Drop} = \frac{Accuracy_{Baseline} - Accuracy_{Fault}}{Accuracy_{Baseline}} \times 100\% \tag{5}$$

The accuracy of varying BERs condition on different layers. Table I reveals the nuanced fault modes of each layer type across six BERs conditions. As for Conv layers, the data collapse rapidly under fault injection, because corrupted filters propagate distorted timing micro-patterns through pooling, irreversibly corrupting spatial-temporal features. Therefore, the accuracy of Conv layers reduce to 29.5% under BER equal to 3×10^{-6}. As for FC layers, the logarithmic accuracy decreases to 1.11% under BER equal to 3×10^{-5} and shows nonlinear fault modes, where unbounded logic distortions overwhelm classification logic. As for all layers, system-level faults exacerbate these issues, and the accuracy reduces to 9.76% under BER equal to 3×10^{-6}, a little bit lowering than configurations excluding LSTM layers (10.7%, Excel.LSTM). Although LSTM layers have inherent robustness, they can still further amplify errors propagated from spatial layers (convolutional and fully connected layers).

As for LSTM layers, exclusive bit-flips in LSTM weights retain 93.3% accuracy under BER equal to 3×10^{-5} (0.02% degradation), reflecting their ability to probabilistically average perturbations over sequential inferences. However, this localized fault masking fails catastrophically when injecting faults into upstream Conv/FC layers, which reveals LSTM's dual role, tolerating self-corruption but propagating external faults.

To overcome this issue, it is crucial to improve the reliability of Conv and FC layers. The experiment results indicate that the fault tolerance capability of the spatial layer is weak, while the LSTM layer exhibits strong robustness. Therefore, it is particularly important to introduce fault-tolerant mechanisms before the LSTM layer, which can not only enhance the reliability of the spatial layer, but also effectively suppress the propagation of faults to the LSTM layer, thereby ensuring the stability of the overall system.

B. Comparative Evaluation of Four Boundary Restriction Methods

The minimal impact of boundary restriction methods on accuracy. As shown in Table II, the accuracy drop of various methods range from 0.00% to 2.78%, and the majority accuracy drop of these methods below 0.86%, which means that these methods have slight influence on accuracy. However, the accuracy drop of Ranger Neuron is relatively high compared to the other methods, which because neuron level activation constraints require fine boundary control to balance the original accuracy and robustness of the model. Ranger's neuron level threshold mechanism only sets boundaries based on the local maximum activation values of limited data. This extreme value statistics based on data segments inevitably filters out some normal activation ranges, resulting in high accuracy drop of the model.

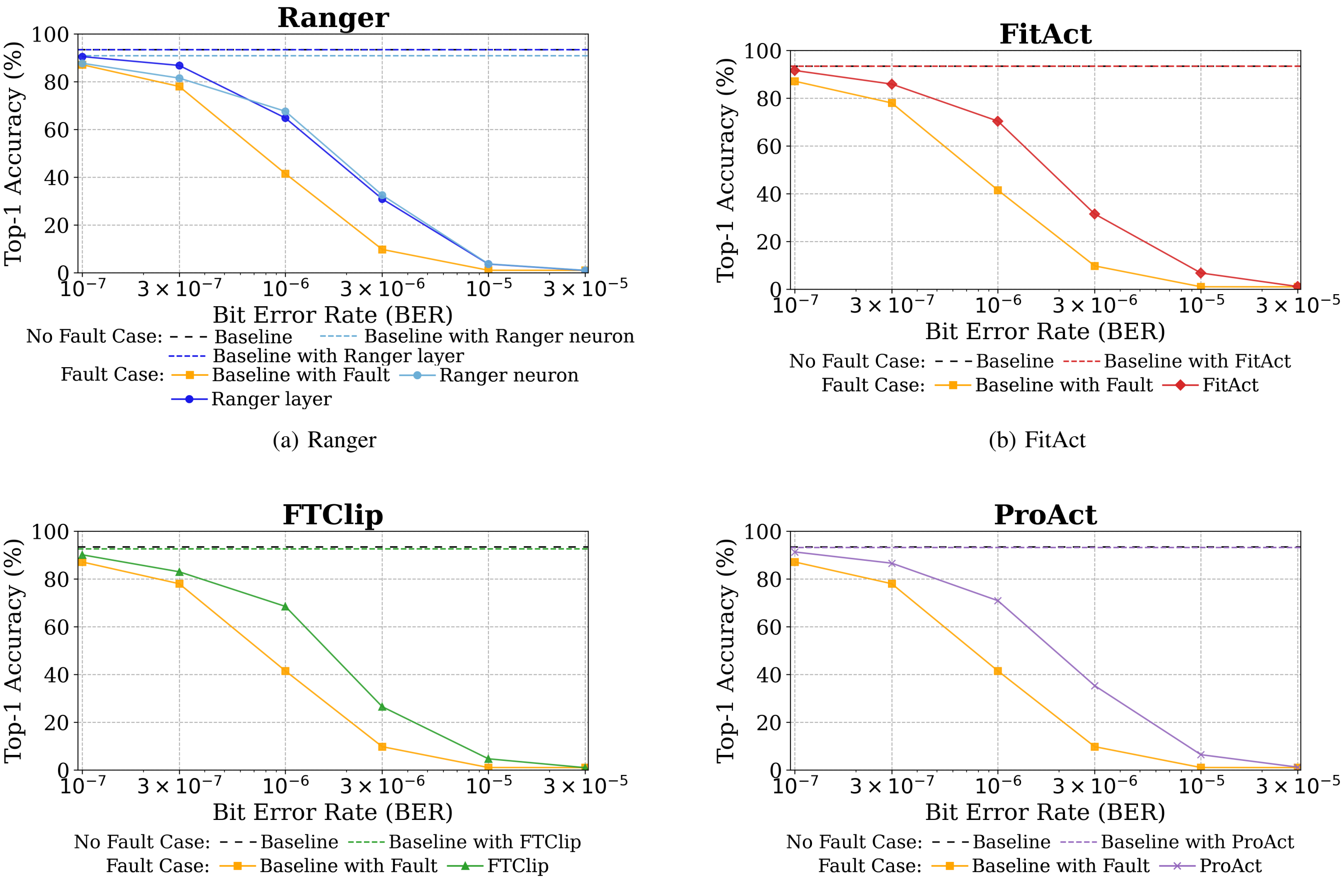
TABLE II: Accuracy Drop of Boundary Restriction Methods in Fault-Free Scenarios

Method	**Accuracy**	**Accuracy Drop**
Baseline	93.5%	-
Ranger Layer	93.5%	0.00%
Ranger Neuron	90.9%	2.78%
FTClip	92.7%	0.86%
FitACT	93.4%	0.11%
ProACT	93.2%	0.32%

TABLE III: The inference time of various boundary restrictions methods

Method	**Inference time**	**Boundary search time**
Baseline	417 μs	-
Ranger layer	526 μs	0.8201 s
Ranger neuron	540 μs	0.8319 s
FTClip	660 μs	2888.1106 s
FitAct	520 μs	590.4113 s
ProAct	508 μs	500.7481 s

The time overhead of boundary restriction methods. The inference time and boundary search time are two key metrics for evaluating the overhead of boundary restriction methods. The former refers to the average prediction time for a single sample, while the latter represents the time required to determine the boundary using different strategies. As shown



(a) Ranger

(b) FitAct

(c) FTClip

(d) ProAct

Fig. 2: Top-1 accuracy comparison of LSTM Model using Ranger neuron, Ranger layer, FitAct, FTClip and ProAct methods under fault injection.

in Table III (Ranger), neuron-level restrictions incur more overhead than layer restrictions, including inference time and boundary search time. This is because neuron-level restrictions require multiple thresholds, whereas layer-level restrictions require only a single threshold. Meanwhile, it is important to note that Ranger has much smaller boundary search time (less than 1s) compared to other methods (range from 500s to 2888s) because it does not include a boundary fine-tuning step and lacks verification of boundary reliability. FTClip (layer-level restrictions) has significantly larger boundary search time (2888s) and inference time growth (58.27%) compared to other methods, due to the different boundary application strategies and boundary search algorithms used by each method. ProACT, a hybrid method, achieves the smallest inference time growth (21.82%) among the five methods and moderate boundary search time, striking a good balance between time overhead and boundary reliability.

The effectiveness of boundary restriction methods across six BERs conditions. As shown in Fig. 2, comparative evaluation of boundary restriction methods reveals distinct trends in balancing accuracy preservation and fault resilience under varying BERs.

The layer restriction methods (Ranger layer, FTClip) suppress fault propagation with negligible baseline accuracy impact (0-0.86%). As shown in Fig. 2a, it can be observed that Ranger is applied at both the layer-level and the neuron-level, and the layer-level restriction is more advantageous at low BERs, because neuron-level restrictions introduce model accuracy loss under fault-free condition when the boundary threshold is unreliable. This accuracy loss gradually decreases when the BERs increase to medium levels, and the accuracy of the layer-level restriction becomes smaller than that of the neuron-level restriction. As shown in Fig. 2c, FTClip, which includes a fine-tuning boundary mechanism, performs similarly to Ranger layer. This is because the limited amount of data affect the effectiveness of boundary fine-tuning, and even inappropriate layer boundary restrictions can also introduce unintended accuracy loss. Furthermore, the uniform layer boundary restriction cannot precisely filter certain faults. Therefore, FTClip can maintain high robustness at low BERs, but requires more refined boundary settings or additional fault mitigation techniques to further improve the model's resilience at high BERs.

The neuron restriction methods (Ranger neuron, FitACT) can filter out anomalous output values better than the layer-level restriction, depending on the different boundary settings of its numerous neurons. As shown in Fig. 2b, the reliability of FitACT is superior to that of Ranger layer and FTClip, achieving the highest accuracy (91.6%) at low BER (10^{-7}), and even maintaining better accuracy than the layer-level restriction at higher BERs. However, the reliability of the method is still unsatisfactory at medium and high BERs. Because neuron-level restrictions require more data and time to validate and fine-tune local thresholds, balancing the model's over-restriction and fault tolerance. Additionally, there is still room for optimization in terms of the inference time and boundary search time at the neuron-level.

The hybrid strategy (ProAct) combines both layer-level and neuron-level restriction. As shown in Fig. 2d, ProAct maintains a minimal accuracy loss without faults (0.3%) while achieving the best reliability. Its hierarchical strategy not only leverages the minimal fault-free accuracy loss from layer-level restrictions and the time overhead advantage compared to neuron-level restrictions under the same method, but also the precise filtering of anomalies by neuron-level restrictions to further improve reliability. This addresses the trade-off between accuracy loss, reliability, and time overhead. Moreover, ProAct performs better than both layer-level and neuron-level constraints alone even with limited data. It achieves 6.1% accuracy improvement over Ranger layer under BER equal to 10^{-6}, and it improves accuracy by 3.8% over FitACT under BER equal to 3×10^{-6}.

In summary, layer-level restrictions are more effective in mitigating faults at lower BERs, while neuron-level restrictions provide better fault resilience at higher BERs. The hybrid restriction (ProAct) demonstrates stable fault tolerance across different BER conditions with relatively low time overhead. If a specific fault mitigation strategy needs to be chosen, the hybrid strategy is the most effective for enhancing the reliability of the CNN-LSTM model. However, all these boundary restriction methods are not particularly reliable under medium and high BERs conditions due to the fact that these methods are only applied to the ReLU activation layers, neglecting the error propagation effect introduced by the LSTM layers. One possible improvement is to extend the hybrid strategy's restrictions to the LSTM layers to further enhance the model's robustness. Moreover, the limited size of the WF model training data (typically $\leq$ 10k trace) and the presence of noise in the data restrict the possibility of a larger performance gap between the methods. As shown in Fig. 2, the different accuracy between the four methods at various BERs conditions is not particularly large. Expanding the dataset size and reducing noise during interrupt data collection could further improve the effectiveness of these methods.

V. Conclusion

This study demonstrates that transient hardware faults significantly impact website fingerprinting attacks by distorting spatial features and inducing error propagation throughout the network. Enhancing the robustness of LSTM-enhanced website fingerprinting models against transient hardware faults requires coordinated spatial-temporal fault containment strategies to mitigate these effects and ensure reliable model performance.

For CNN-LSTM hybrid architectures, our layer-level fault analysis reveals that Conventional and Full-Connected layers contribute severely to accuracy degradation, whereas isolated LSTM faults only marginally impact performance. However, simultaneous faults across all three components amplify distortions, causing a cascading accuracy decline. Unlike many other DNN applications—such as image recognition or natural language processing—where fault impacts can often be absorbed by network redundancies or offset by retraining, the unique reliance of website fingerprinting on minute temporal patterns renders it especially vulnerable to even minimal hardware perturbations. This distinct sensitivity necessitates tailored fault mitigation strategies that not only address spatial error propagation but also safeguard the integrity of critical temporal features.

Conventional layer-level methods (e.g., Ranger Layer) suffer catastrophic failures under high BERs, dropping to 30.9% accuracy under BER equal to 3×10^{-6}, while neuron-level approaches sacrifice significant accuracy in fault-free scenarios (e.g., 2.6% loss for Ranger Neuron) for marginal resilience. The hybrid strategy (ProAct) strikes a balance between reliability and fault-free accuracy loss, incorporating both layer-level and neuron-level constraints to effectively suppress fault propagation across spatial layers, while also offering relatively lower time overhead compared to other methods. Under a BER of 3×10^{-6}, it achieves an accuracy of 35.3%, with only a 0.3% decrease in fault-free accuracy (93.2% compared to the original 93.5%). Critically, this evaluation reveals that transient fault resilience in hybrid architectures depends on targeted suppression of spatial error propagation to protect temporal feature dynamics, a principle essential for maintaining the accuracy of website fingerprinting attacks under hardware-driven perturbations. Hybrid strategy (ProAct) demonstrates that selectively restricting spatial layers can protect temporal feature dynamics, thereby improving attack reliability in fault-prone scenarios. However, significant accuracy degradation persists, necessitating further extension of the mitigation strategy into LSTM layers to restrict the amplification of faults.

This work underscores the need for cross-layer fault modeling in privacy-sensitive machine learning. Future research should prioritize two directions: 1) Standardized interrupt-driven benchmarks to address dataset scale limitations and improve threshold calibration across fault scenarios, and 2) Further extend the mixed strategy method in boundary restriction to the LSTM, and appropriately restrict the LSTM layer to pursue the minimum accuracy degradation. Such advancements will strengthen the co-design of reliability and robustness in WF systems operating in unpredictable hardware environments.

References

[1] A. Voulodimos, N. Doulamis, A. Doulamis, and E. Protopapadakis, "Deep learning for computer vision: A brief review," Computational Intelligence and Neuroscience, vol. 2018, no. 1, p. 7068349, 2018.

[2] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," Briefings in Bioinformatics, vol. 19, no. 6, pp. 1236–1246, November 2018.

[3] C. Torres-Huitzil and B. Girau, "Fault and error tolerance in neural networks: A review," IEEE Access, vol. 5, pp. 17322–17341, 2017.

[4] N. Narayanan, "Fault injection in Machine Learning applications," Ph.D. dissertation, University of British Columbia, 2021.

[5] M. Stoffel, M. Schindewolf, and E. Sax, "On-Demand Triple Modular Redundancy for Automotive Applications," in 2024 IEEE International Systems Conference (SysCon). IEEE, 2024, pp. 1–8.

[6] S. S. Liew, M. Khalil-Hani, and R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems," Neurocomputing, vol. 216, pp. 718–734, 2016.

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

[8] P. Liu, L. He, and Z. Li, "A survey on deep learning for website fingerprinting attacks and defenses," IEEE Access, vol. 11, pp. 26033–26047, March 6, 2023.

[9] S. Lau, T. Bourgeat, C. Pit-Claudel, and A. Chlipala, "Specification and Verification of Strong Timing Isolation of Hardware Enclaves," in Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security, December 2024, pp. 1121–1135.

[10] A. Mahmoud, N. Aggarwal, A. Nobbe, J. R. S. Vicarte, S. V. Adve, C. W. Fletcher, . . . , and S. K. S. Hari, "Pytorchfi: A runtime perturbation tool for DNNs," in 2020 50th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W), June 2020, pp. 25–31. IEEE.

[11] Z. Chen, G. Li, and K. Pattabiraman, "A low-cost fault corrector for deep neural networks through range restriction," in 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). IEEE, 2021.

[12] L. H. Hoang, M. A. Hanif, and M. Shafique, "FT-ClipAct: Resilience analysis of deep neural networks and improving their fault tolerance using clipped activation," in 2020 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2020, pp. 1241–1246.

[13] B. Ghavami, M. Sadati, Z. Fang, et al., "FitAct: Error resilient deep neural networks via fine-grained post-trainable activation functions," in 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2022, pp. 1239–1244.

[14] S. Mousavi, et al., "ProAct: Progressive Training for Hybrid Clipped Activation Function to Enhance Resilience of DNNs," arXiv preprint arXiv:2406.06313, 2024.

[15] G. Li, K. Pattabiraman, and N. DeBardeleben, "TensorFI: A configurable fault injector for TensorFlow applications," in 2018 IEEE International Symposium on Software Reliability Engineering Workshops (ISSREW), IEEE, 2018, pp. 313–320.

[16] S. E. Oh, T. Yang, N. Mathews, J. K. Holland, M. S. Rahman, N. Hopper, and M. Wright, "DeepCoFFEA: Improved flow correlation attacks on Tor via metric learning and amplification," in 2022 IEEE Symposium on Security and Privacy (SP), May 2022, pp. 1915–1932. IEEE.

[17] J. Cook, J. Drean, J. Behrens, and M. Yan, "There's always a bigger fish: A clarifying analysis of a machine-learning-assisted side-channel attack," in Proceedings of the 49th Annual International Symposium on Computer Architecture, June 2022, pp. 204–217.

[18] G. Li, S. K. S. Hari, M. Sullivan, T. Tsai, K. Pattabiraman, J. Emer, and S. W. Keckler, "Understanding error propagation in deep learning neural network (DNN) accelerators and applications," in Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, November 2017, pp. 1–12.

[19] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint arXiv:1803.01271, 2018.

[20] A. Shusterman, L. Kang, Y. Haskal, Y. Meltser, P. Mittal, Y. Oren, and Y. Yarom, "Robust Website Fingerprinting Through the Cache Occupancy Channel," CoRR, vol. abs/1811.07153, 2018, arXiv preprint arXiv:1811.07153.

[21] A. Shusterman, A. Agarwal, S. O'Connell, D. Genkin, Y. Oren, and Y. Yarom, "Prime+Probe 1, JavaScript 0: Overcoming Browser-based Side-Channel Defenses," in 30th USENIX Security Symposium (USENIX Security 21), 2021, pp. 2863–2880.

[22] O. AL-Jarrah and A. Arafat, "Network intrusion detection system using neural network classification of attack behavior," Journal of Advances in Information Technology, vol. 6, no. 1, 2015.