Vol.4, No.8, Page 2292- 2301, 2026 https://doi.org/10.59628/jast.v4i8.3272 ISSN 2958-9568 JAST

SUN PUBLISHER

مجلة جامعة صنعاء للعلوم التطبيقية والتكنولوجيا

Sana'a University Journal of Applied Sciences and Technology

https://journals.su.edu.ye/index.php/jast/

A Comparative Analysis of Feature Representation Paradigms for Website Fingerprinting on the Java Anon Proxy Network

Abdulqader Shaawa [1, 2 *] **and Abdoulwase Al Azzani** [1]

[1] **Department of Computer Science, Faculty of Computer and Information Technology, Sana'a University, Sana'a, Yemen,**
[2] **Department of Computer Science, Faculty of Computer Engineering and Science, Hodeida University, Hodeida, Yemen**

*Corresponding author: kader.shawa@su.edu.ye

Abstract

Website fingerprinting (WF) attacks pose a serious threat to online privacy, even when users employ anonymizing networks. The effectiveness of deep learning-based WF critically depends on how raw network telemetry is transformed into input representations. In this paper, we systematically compare five feature encoding paradigms–Direction Only, Inter-arrival Times, Unsigned Packet Sizes, Signed Inter-arrival Times, and Signed Packet Sizes–using two deep learning architectures: a 1D Convolutional Neural Network (CNN) and a CNN-LSTM-Attention (CLA) model. We conduct our experiments on a newly collected Java Anon Proxy (JAP) dataset of 10,000 traces, measuring accuracy, stability (standard deviation and learning curves), training time, and peak RAM usage. Our results show that Direction Only achieves high accuracy (99.14% with CLA) and low variance, confirming its discriminative power. However, Signed Packet Sizes–combining packet size with directional sign–deliver the best overall performance: CLA reaches 99.56% accuracy with low variance (0.24%), while the CNN model attains 98.22% with even lower variance (0.18%) and faster training time. A sensitivity analysis with four weighting schemes (accuracy-focused, efficiency-focused, stability-focused, balanced) shows that while Signed Sizes-CLA is optimal for absolute peak accuracy, Signed Sizes-CNN provides the best trade-off for practical deployments where computational resources are limited.

Article Info

Keywords:
Website fingerprinting, JAP, Anonymity, WF attack, Deep learning

Article History:
Received: 31-May-2026,
Revised: 21-June-2026,
Accepted: 9-July-2026,
Published: 28 August 2026.

1. INTRODUCTION

Protecting privacy is one of the most significant challenges users face in today's digital age. Many entities–such as major social media platforms, advertising agencies, internet service providers, and even some repressive regimes–violate users' privacy, regardless of their motives. Although encrypted communications are widespread, they do not provide complete protection, as they are limited to protecting the data being exchanged. It does not protect the identity of the communicating parties. Consequently, an eavesdropper can obtain essential information without needing to decrypt the communication data. For example, attackers can deduce the identity of the parties involved (i.e., the website the user is visiting), the timing of the connection, its duration, and the volume of data exchanged.

In response, users have turned to anonymization services such as Onion Routing (Tor) [1], Java Anon Proxy (JAP) [2, 3], Invisible Internet Project (I2P) [4], and virtual private networks (VPNs) to hide their online activities and ensure their privacy. In turn, attackers have developed a new type of attack based on data analysis, even when these services are used. Among these attacks, website fingerprinting has emerged as a serious threat, as it enables the identification of websites visited by users by analyzing patterns of encrypted traffic. This method typically relies on classifying websites by extracting distinctive traffic characteristics, which are then classified

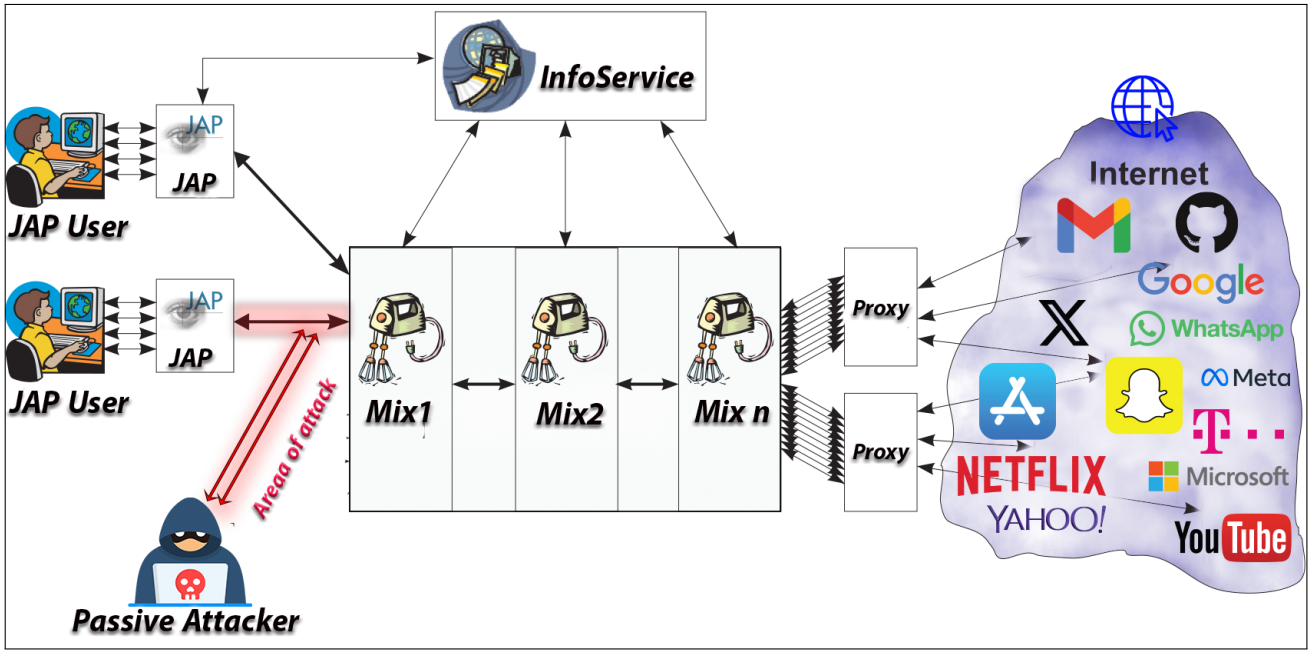


Figure 1. Website Fingerprinting (WF) attack scenario within the JAP architecture.

using various machine learning algorithms [5] [6] [7].

As illustrated in the JAP-centric threat model in Figure 1, the adversary monitors the connection between the JAP client and the server providing the anonymization service. The attacker captures encrypted traffic data as the victim visits various websites. He then compares this data against a library of metadata "fingerprints"–such as the size, direction, and timing of exchanged packets–that he has built using an environment that mimics the victim's. By training machine learning models on this library, the attacker can identify which websites the user is visiting, thereby fundamentally compromising the privacy of their browsing.

The remainder of this paper is organized as follows. Section 2 provides a background and reviews related work. Section 3 describes our methodology and experimental framework. Section 4 reports and discusses the results. Finally, Section 5 concludes the paper.

2. BACKGROUND AND RELATED WORK

Since early in the new millennium, website fingerprinting attacks–targeting anonymization services–have seen a remarkable evolution, met with relentless efforts to develop countermeasures to limit their success. At a certain stage, these attacks become a classification process whose success is dependent on the algorithm and the data used in the training process. The proposed defenses demonstrated the potential to limit the success of these attacks by obscuring the distinctive features that can be extracted from the data, even though it is encrypted.

In this section, we classify and summarize previous work on WF attacks and the features used in them; subsequently, we provide the necessary background on the models used, in addition to JAP as an example of anonymization services.

2.1. Evolution of WF Attacks

Early Foundations: Yee's observation regarding the relationship between the length of the ciphertext and the plaintext–in encryption systems that do not rely on random padding–formed the basis for all subsequent research on fingerprinting attacks [8]. Building on this, Wagner and Schneier highlighted the possibility that an observer–by analyzing encrypted data traffic–could identify the web pages accessed by SSL/TLS (Secure Sockets Layer / Transport Layer Security) [9].

In the early 2000s, Cheng et al. [10] were able to identify pages accessed via SSL by analyzing the number and sizes of requested page objects. The term "website fingerprint" is attributed to "Hintz", who published his attack against the SaveWeb proxy [5]. Simultaneously, Sun et al. – referring to the same technique as "traffic signature" – achieved a 75% detection rate using Jaccard's coefficient as a similarity metric [11].

The common problem with all the aforementioned attacks achieve is that they are based on the assumption that every web element passes through a separate TCP connection. Although this assumption is valid in HTTP 1.0, it is no longer valid for newer HTTP 1.1 protocols. Furthermore, this assumption cannot be successfully applied against tunnel-based privacy-enhancing technologies (such as VPNs, Open Secure Shell(OpenSSH), Tor, and JAP) that hide individual connections (and thus the sizes of the elements) from eavesdroppers. Consequently, the aforementioned techniques that rely on the

number and sizes of requested objects are no longer effective.

2.1.1. Classical Machine Learning and Statistical Approaches

The use of machine learning models marked a turning point in WF attacks and a significant paradigm shift. Bissias et al. led this shift from element-based analysis to statistical features extracted from encrypted data, such as packet sizes and timings, achieving 23% accuracy (100% when allowing three guesses) against OpenSSH using cross-correlation [7]. This work opened the door to exploiting temporal information alongside packet sizes in subsequent research. Liberatore and Levine [6] and Herrmann et al. [12] further advanced this approach by applying Naïve Bayes classifiers (NB) to packet size distributions.

With the emergence of more advanced algorithms–such as K-Nearest Neighbors (KNN), Random Forest (RF), and Support Vector Machine (SVM)–the success rate of these attacks exceeded the 90% threshold, demonstrating a significant threat and highlighting the vulnerability of more advanced anonymization systems like JAP and Tor [13] [14] [15]. However, the success of these attacks depended on the attacker's expertise in manually extracting features from raw data, and these methods failed to generalize their success across different conditions or networks.

2.1.2. Deep Learning Revolution

Due to the success of deep learning models in image recognition, many researchers have adopted these models in the field of website fingerprinting [16, 17]. These models outperformed their traditional machine learning counterparts and addressed all the shortcomings mentioned above. Although requiring significant data and computational resources for training, they have achieved a higher success rate.

Abe and Goto (2016) were the first to implement an attack using a Stacked Denoising Autoencoders (SDAE) on Tor traffic, achieving an accuracy rate of 88%. Although this was lower than what had been achieved in ML, their work demonstrated the potential of deep learning models for automatic feature extraction [18]. Later, Rimmer et al. (2018) systematically compared SDAE, CNN, and Long Short-Term Memory (LSTM) on a large dataset (900 sites, 2,500 traces per site). Their CNN model reached 96% accuracy, confirming the superiority of standard deep learning models over classical models [19]. On the other hand, Sirinam et al. (2018) designed a Deep Fingerprinting (DF) with batch normalization and dropout, achieving 98% accuracy. Their use of regularization techniques was crucial for demonstrating the absence of overfitting and formed the basis for most subsequent work [20].

Subsequent works focused on improving the efficiency of these attacks. Bhat et al. (2019) introduced Var-CNN, which uses ResNet-style connections – or skip connections, allow deep neural networks to bypass intermediate layers, enabling gradients to flow unimpeded during training– to handle variable-length traces with fewer training samples. Similarly, Sirinam et al. (2019) proposed triplet fingerprinting, achieving 95% accuracy with only 20 examples per website [21] [22].

Oh et al. (2020) used Generative Adversarial Networks (GANDaLF) to generate training data when real traces are scarce. Later, Lu et al. (2021) introduced Graph Attention Pooling Network for Fine-grained Website Fingerprinting (GAP-WF), that models web page loading as a graph of traces, reducing computational complexity while maintaining high accuracy [23] [24]. In addition to adopting more realistic scenarios such as multi-tab browsing, recent research has focused on proposing low overhead defenses and evaluating their effectiveness against enhanced attacks [25] [26].

Table 1 summarizes the evolution of WF attack methodologies, highlighting key architectural innovations and reported performance. It is clear that recent research has focused on the Tor network at the expense of other anonymity services. As the world's most widely used anonymization system, it has become the primary battleground for location fingerprinting attacks and their defenses. In contrast, other privacy-enhancing technologies, such as JAP, remain critically underexplored. Although JAP introduces a model based on cascaded mixes that differs from Tor's dynamic circuit-based routing model, no prior study has systematically compared the impact of different JAP traffic features on the latest generation of deep learning-based website fingerprinting (WF) attacks. Moreover, we go beyond simple accuracy comparison by incorporating stability (variance and learning curves), and computational efficiency (training time, RAM).

2.2. The AN.ON (JAP) Anonymization System

JAP (Java Anonymous Proxy), also known as JonDo in the commercial variant, was originally developed as a project of the Technical University Dresden and the Independent Center for Data Privacy Schleswig-Holstein. It allows users to surf the Internet anonymously and unobservably by hiding their real IP address behind a single static address shared by many JAP users [3]. The structure of JAP consists of four main components:

- **JAP client:** A Java application installed locally that connects to the first MIX. It registers the user, generates an asymmetrically encrypted channel, and filters anonymity-threatening content (e.g., cookies, JavaScript).
- **MIXes:** Simple computers forming a cascade. Each

Table 1. Evolution of website fingerprinting attack methodologies (expanded).

Year	Authors	Method	Target	Accuracy
2002	Sun et al.	Jaccard's coefficient	SSL proxy	75%
2006	Liberatore & Levine	NB + Jaccard	OpenSSH	∼70%
2009	Herrmann et al.	Multinomial NB	JAP, Tor	20% (JAP), 2.95% (Tor)
2011	Panchenko et al.	SVM	Tor, JAP	>90%
2013	Wang & Goldberg	SVM + OSAD / FLeV	Tor	94% (OSAD)
2014	Wang et al.	KNN	Tor	95%
2016	Hayes & Danezis	Random forest	Tor	85-95%
2018	Rimmer et al.	SDAE, CNN, LSTM	Tor	96% (CNN)
2018	Sirinam et al.	CNN (DF)	Tor	98%
2019	Sirinam et al.	Triplet networks	Tor	95% (85% with 5-shot)
2020	Oh et al.	GANDaLF	Tor	87% 20 instance per site
2021	Lu et al.	GAP-WF	SSL/TLS	99.86%
2025	Yuan et al.	Multi-tab WF	Tor	90%

MIX takes a number of messages, mixes them by encryption/decryption, reordering, and delaying, then outputs them in random order. This prevents correlating incoming and outgoing messages. JAP implements a modified MIX system originally invented by Chaum [27].

- **Cache proxy:** Connected to the last MIX, it forwards data to the Internet and relays responses back through the MIX cascade.
- **InfoService:** Provides meta-information about available MIX cascades (addresses, public keys, user counts) to determine the anonymity level.

When a user starts JAP, the client retrieves meta-information from the InfoService, registers with the first MIX, and then encrypts and sends data through the cascade. Each MIX applies cryptographic transformations, and the last MIX delivers the data via the cache proxy to the destination. Responses travel back in reverse order.

3. METHODOLOGY

The success of both traditional and deep learning models depends on the effective combination of the model and data quality used in the training process. While the success of classical learning models relies on the attacker's expertise in extracting and preprocessing data, deep learning models have surpassed them in their ability to automatically extract distinctive features. This justifies the focus of the vast majority of proposed defenses against website fingerprinting attacks on obscuring distinctive features in data streams, despite the trade-offs in terms of data size and latency.

Our main objective in this study is to conduct an experimental comparison to select the appropriate features for deep learning models. We did not limit our focus to the combination that achieves the highest quality; rather, we expanded the comparison to include the stability of the learning process and its efficiency in terms of training time and resource consumption.

3.1. Dataset and Preprocessing

In our experiments, we relied on JAP data recently collected by Shaawa and Al-Azzani [26]. These websites were selected based on several earlier studies to enable direct comparison with prior Tor-based attacks [6] [12] [14]. This sample includes a mix of high-traffic sites from the Alexa Top 500, as well as specific categories (such as news, social media, and finance) to ensure diversity in traffic patterns.

Automated Traffic Capture: browsing and data collection process automated using Selenium WebDriver, and a Python script. The script sequentially visits each of the 100 selected URLs through a locally configured JAP client. For each visit, the script simultaneously captures all network traffic and saves it as a separate PCAP file named '[URL]_[timestamp].pcap', establishing a clear one-to-one mapping between a traffic trace and its website label. As a result of this automated process, the final raw dataset was composed of 10,000 PCAP files (100 websites * 100 instances per site), which served as the foundation for all subsequent analysis.

Using the dpkt library, each PCAP file was parsed to extract three key attributes from every packet: size (in bytes), direction relative to the client (+1 for outgoing, -1 for incoming), and timestamp. This step results in three lists that represent each of the key features, as shown in Eq. (1):

$$\begin{aligned}\text{packet\_sizes} &= [1064,\ 66,\ 1064,\ 66,\ 1454,\ \ldots]\\ \text{packet\_directions} &= [1,\ -1,\ -1,\ -1,\ -1,\ \ldots]\\ \text{packet\_times} &= [0.0,\ 0.1935,\ 0.2119,\ 0.4409,\ 0.4410,\ \ldots]\end{aligned} \tag{1}$$

Based on this raw data, numerous features can be extracted and used as inputs for classical and deep learning models. For example, the total number of packets, total volume, etc., can be extracted based solely on packet sizes. Direction information–which is a binary sequence where incoming packets are represented by +1 and outgoing packets by -1–is also a rich source of features, such as the number of incoming and outgoing packets. When direction is combined with size data, the size and

Table 2. Taxonomy of traffic features utilized in the comparative evaluation (adapted from [28]).

Feature Category	Descriptors	Experimental Utility
Timing	Inter-packet intervals, RTT, quiescent periods	Capture request–response temporal patterns
Length	Aggregate bytes, packet counts, directional volume	Evaluate coarse-grained volumetric footprints
Packet Size	MTU, non-MTU packets, unique dimensions	Identify resource-specific magnitudes
Packet Ordering	Egress/ingress sequences, request order	Differentiate domains with fixed asset loading orders
Bursts	Magnitude, frequency, and position of bursts	Encapsulate high-level structural hierarchy
Direction	Egress-to-ingress ratios, directional transitions	Distinguish upload-centric from download-centric traffic
Coarse-grained	Throughput, trace duration, stochasticity	Facilitate rapid filtering and low-resource analysis

number of incoming and outgoing packets can be calculated. In addition, packet sequences and the number of transitions between sending and receiving are considered important features in the literature. Through these, it is possible to distinguish between text-based interactive websites–where incoming and outgoing data alternate frequently–and media-based websites, which are dominated by a larger volume of incoming data. The third vector in Eq. (1) represents the temporal patterns of packets, such as the time interval between their arrival and the total time required to fully load the website. Table 2, adopted from [28], illustrates the most common features based on the above fundamental vectors.

3.2. Taxonomy of Traffic Features

The efficacy of any Website Fingerprinting (WF) exploit is fundamentally predicated upon the observability and discriminative capacity of specific traffic features. Adopting a systematic categorization based on the structural properties of network traffic [28], we identify seven primary groups of descriptors: timing, length, packet dimensions, ordering, bursts, directionality, and coarse-grained statistics. Table 2 summarizes these categories and their respective roles within our experimental evaluation.

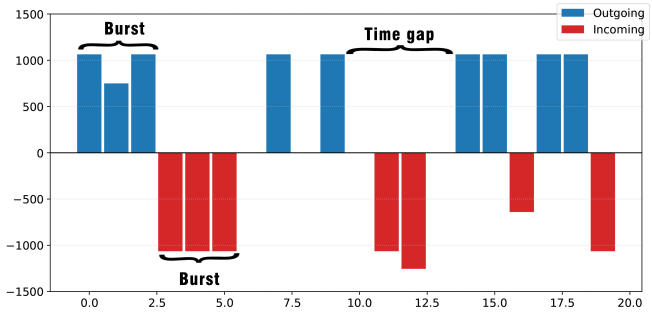


Figure 2. Common features of JAP traffic(direction, size, burst, ordering, and time gap).

To ascertain the optimal input representation based on the taxonomy described above, we synthesized the raw telemetry into the following five sequential encodings:

i. **Directional Only:** A binary sequence isolating flow control patterns (+1 for egress, -1 for ingress).
ii. **Inter-Packet Intervals (IPI):** A scalar sequence focusing exclusively on temporal inter-arrival signatures. The first packet has no preceding interval, so its value is 0.
iii. **Unsigned Packet Sizes:** A volumetric sequence evaluating the discriminative power of packet magnitudes devoid of directional context.
iv. **Signed Inter-Packet Intervals:** A hybrid temporal-directional representation where the magnitude denotes the inter-arrival time and the sign denotes the direction of the packet that arrived after that interval (the first element is 0, similar to IPI attacks).
v. **Proposed Signed Packet Sizes:** A consolidated representation where magnitude denotes packet volume and the sign signifies directionality.

Figure 2, illustrates these common features of JAP traffic(direction, size, burst and time gap).

3.3. Deep Learning Models

In our work, we mainly focus on two deep learning techniques that previous work has shown to be promising for WF attacks. We used two architectures:

- **1D Convolutional Neural Network (1D-CNN):** The 1D-CNN model treats the packet sequence as a temporal signal and applies convolutional filters to identify local patterns within the traffic trace. This model can learn local patterns of bursts and packet sizes to identify websites through fingerprinting.
- **CNN-LSTM-Attention (CLA):** It combines the strengths of CNN's capability to extract local patterns with the ability of LSTMs to recognize long-term dependencies in both directions. The role of the attention layer here is to weigh the input packet's data dynamically based on its significance in the classification process. This mechanism enables the model to focus on the most informative traffic segments,

thereby improving classification accuracy and accelerating training convergence.

Table. 3 summarizes key hyper-parameters and configurations of the deep learning models used in our experiments, including the optimizer, learning rate, batch size, number of epochs, sequence length, and the hardware environment used for training.

Table 3. Deep learning model configurations and training details.

Parameter	Value
Optimizer	Adam
Learning rate	0.001
Batch size	64
Epochs	30
Sequence length	3000 (padded/truncated)
Dropout rate	0.3
Batch Normalization	Yes
Hardware	NVIDIA Tesla T4 (16GB)
Framework	Keras / TensorFlow 2

3.4. Training and Cross Validation

We performed 5-fold stratified cross-validation (80/20 split per fold) with a fixed random seed (42). Each fold was trained for up to 30 epochs with early stopping (patience=5 on validation loss and learning rate reduction on plateau (factor 0.5, patience=3)). Training and validation accuracy/loss were recorded per epoch for each fold. Resource usage (training time per fold, peak RAM) was measured and logged, to ensure computational overhead remained within manageable limits throughout the iterative learning process.

4. Results and Analytical Discussion

Using both the dataset and the successful models mentioned in the previous section, we applied these attacks but with a different distinguishing feature each time. We started by using the three basic features from the Eq. (1) individually and then expanded to combine the volume and timing features with the directional feature to obtain the best combination (model + features) not only in terms of attack accuracy but also in terms of attack efficiency.

The results in Table 4, particularly the first three features, compare the performance of each model used against each individual feature. We can clearly see that the direction feature outperforms the size and timing features, achieving an accuracy rate of 99.14% with the CLA model. The CNN model, also using the direction feature, performed well with an accuracy of 97.5%, but with a shorter training time. The small standard deviation reflects the stability of the training process for both models, highlighting the importance of this feature and justifying its widespread use in previous research.

When it comes to size and timing features, these models achieved lower accuracy compared to the direction feature. Although they require less training time, they suffer from low accuracy and instability.

When features are combined, the model can leverage this and extract more information. In addition to the distinctive features provided by direction information–such as the number of incoming and outgoing packets, as well as bursts–combining them with other features such as size and timing provides additional information that improves the classifier's learning process. For example, when combining direction with size in a single vector, as shown in Eq. (2), the model benefits from additional information such as the total data volume and the volume of incoming and outgoing data. Similarly, combining timing data with direction –Eq. (3)– represents a clear improvement compared to timing features alone, as shown in Table 4 above.

$$\text{signed\_packet\_sizes} = [1064,\ -1064,\ -1454,\ \ldots] \quad (2)$$

$$\text{signed\_interval\_sizes} = [0.0,\ -0.1935,\ -0.2119,\ \ldots] \quad (3)$$

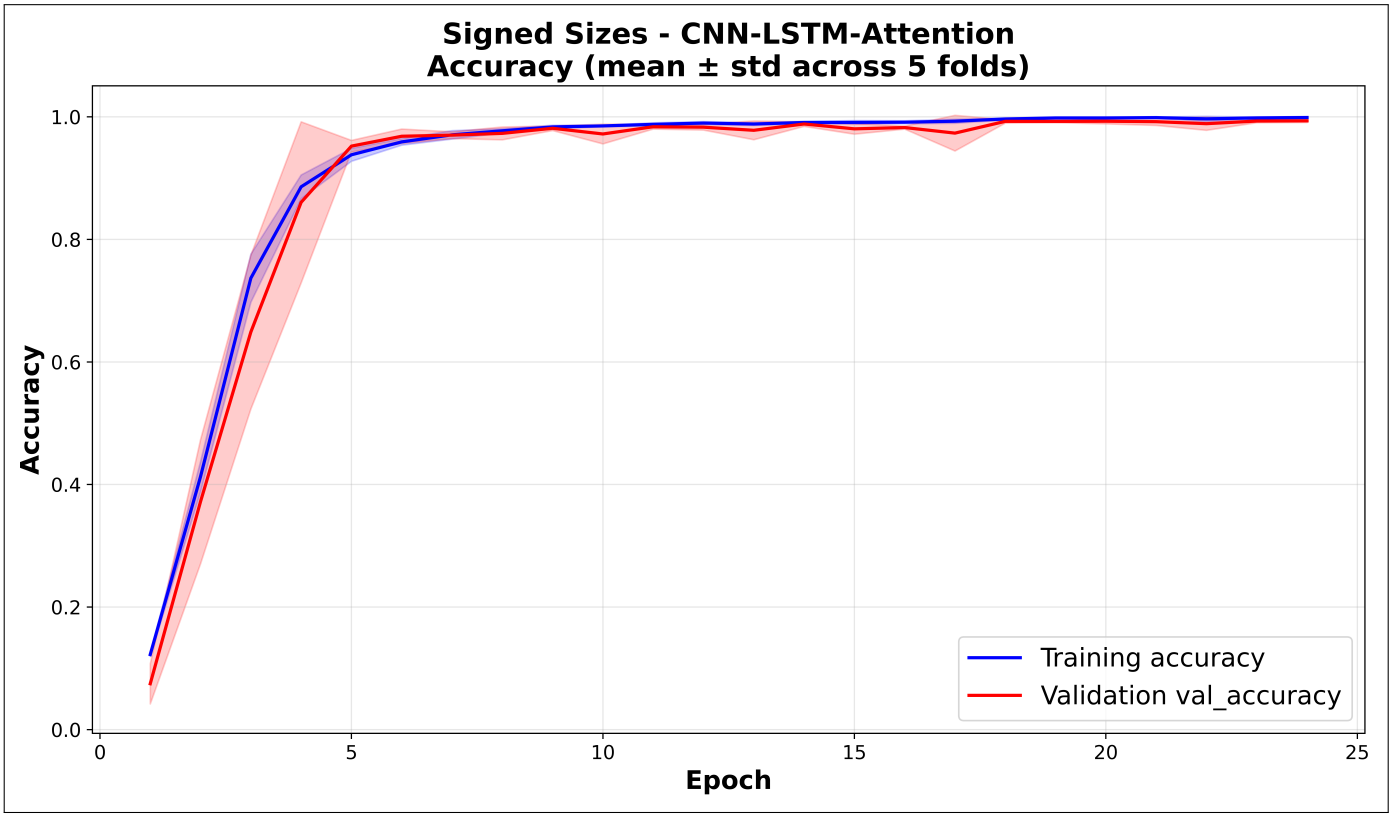
As shown in Table 4, the additional features obtained by combining size and direction are more influential than those resulting from the addition of timing data. The signed sizes with the CLA model achieved the highest absolute accuracy of 99.56%, compared to 98.22% achieved by the CNN model, which was the most stable among the models. While combining raw timing information with directional features did not add value compared to directional features alone, it performed better than timing features alone.

4.1. Resource Consumption and Stability

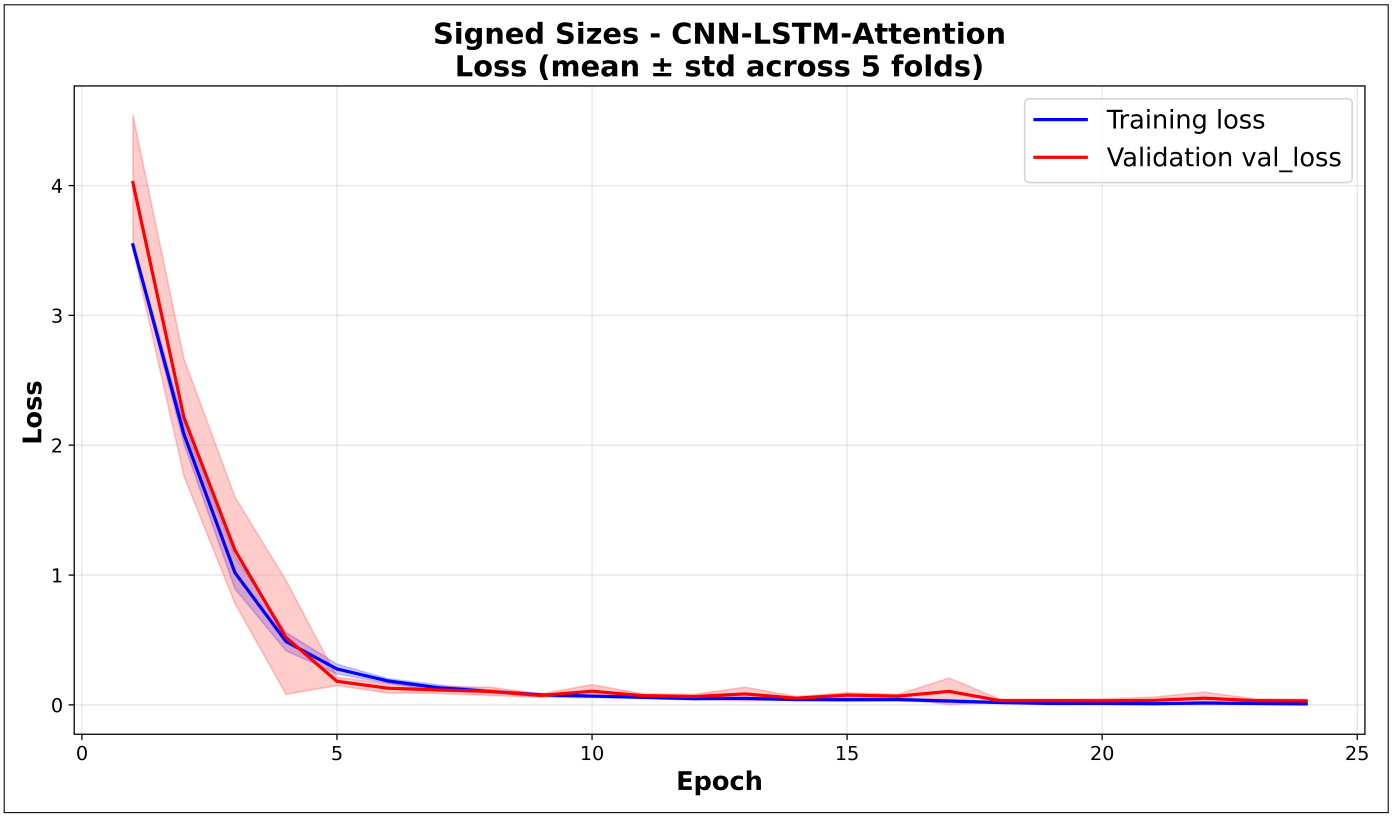
It can be observed that the **training time** for the CLA model is 3–4 times longer in comparison to the CNN model for all features used. This discrepancy in computational latency stems from the inherent structural complexity of the attention-augmented LSTM layers, which necessitates sequential processing and multiple weight updates across time steps, whereas the 1D-CNN utilizes parallelizable convolution operations that significantly reduce overhead.

On the other hand, single features such as size and timing are relatively faster than composite features but with lower accuracy, with the exception of the direction feature, which achieved 97.5% accuracy in just 20 minutes using the CNN model.

Peak RAM: For most features, CLA consumed slightly less or the comparable amount of RAM compared to CNN. For example, CLA used only 2,470 MB for direc-



(a) Training and validation accuracy of the CLA model using signed packet sizes. The narrow shaded region indicates high stability and absence of overfitting.



(b) Training and validation loss of the CLA model using signed packet sizes. Both curves converge smoothly to low values, confirming consistent learning.

Table 4. Empirical Performance Comparison of Individual Feature

Feature	Model	Avg Acc (%)	Std Acc (%)	TrainTime (min)	Peak RAM (MB)
Direction Only	CNN	97.58	0.25	20.5	3425
	CLA*	99.14	0.26	84.3	2470
Inter-arrival	CNN	73.91	36.36	17	3406
	CLA*	88.60	9.93	52	2429
Unsigned Size	CNN	92.85	0.70	17	3120
	CLA*	82.23	14.71	60	2439
Signed Interval	CNN	95.21	0.39	29	3364
	CLA*	97.75	1.37	89	2429
Signed Sizes	CNN	98.22	0.18	22	3077
	CLA*	99.56	0.24	74	3234

*CLA denotes the CNN-LSTM-Attention architecture.

Table 5. Feature performance trade offs.

Priority	Top ranked combination	Key traits
Accuracy focused	Signed Sizes – CLA	Highest accuracy (99.56%)
Efficiency focused	Inter arrival – CNN	Fastest but low accuracy (73.91%)
Stability focused	Signed Sizes – CNN	Lowest variance (0.18%)
Balanced	Signed Sizes – CNN	High accuracy (98.22%), low variance, fast

tions, compared to 3,425 MB for CNN. However, for signed sizes, CLA required 3,234 MB, which is approximately 5% more than CNN (3,077 MB). This exception is due to the increased number of relevant features retained by the attention mechanism, which is associated with the highest accuracy achieved by this combination.

Training **Stability** (low standard deviation) is considered one of the most important indicators of the absence of overfitting, as the gap between the training and validation accuracy curves indicates strong generalization ability. Our results showed that models based on the combination of direction and size features exhibited lower variance (standard deviation) during the training and validation phases. Figure 3a shows that the CLA model achieves an accuracy of over 99% within 10 training epochs, after which improvements become minimal. Figure 3b illustrates a smooth and steady decrease in loss throughout the training period, with both training and validation losses converging to low values as improvements stabilize.

4.2. Trade offs in Practice

As shown in Table 5, the balanced scenario places Signed Sizes – CNN as the most robust overall, offering near-optimal accuracy with significantly lower computational cost than its CLA counterpart. For applications where absolute peak accuracy is the sole requirement, Signed Sizes—CLA is preferred.

Our results align with deep learning WF literature. Sirinam et al. [20] reported 98% accuracy on undefended Tor using Deep Fingerprinting; our Signed Sizes CLA achieves 99.56% on JAP, showing that JAP is at least as vulnerable. The failure of unsigned sizes with attention echoes observations that direction-aware inputs are critical for recurrent architectures. To our knowledge, this is the first systematic comparison of five feature encodings for JAP, providing empirical justification for signed packet sizes as the optimal representation.

5. CONCLUSION

This paper systematically evaluated five feature encoding paradigms for website fingerprinting (WF) on Java Anonymous Proxy (JAP) traffic: direction only, inter-arrival times, unsigned packet sizes, signed inter-arrival times, and signed packet sizes. Using two deep learning models (CNN and CNN-LSTM-Attention), we conducted 5-fold cross-validation and measured accuracy, stability, training time, and memory usage.

Our results show that the direction feature alone is a powerful feature, achieving an accuracy of 99.14% with the CLA model, confirming that directional sequences capture fundamental request-response patterns. Packet sizes alone perform reasonably well with a CNN (92.85%) but fail with the CLA model (82.23%, high variance). This failure is due to the fact that the attention mechanism performs better when combining directional context with size information.

Signed packet sizes provide the best overall performance: the CLA achieves an accuracy of 99.56% (low variance), while the CNN achieves an accuracy of 98.22% with lower variance (0.18%) and 3.4 times faster training.

In general, if the goal is to achieve the highest possible accuracy, the CLA model with signed sizes is the optimal choice. However, when computational resources are limited, the CNN model with signed sizes offers the best balance: near-perfect accuracy, optimal stability, and fast training.

We plan to evaluate the robustness of signed size representations against existing WF defenses (e.g., WTF PAD, RBCT) and to explore multi modal fusion (e.g., concatenating signed sizes with timing features). Ad-

ditionally, transferability of the learned features across different network conditions warrants investigation.

REFERENCES

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router:" Defense Technical Information Center, Fort Belvoir, VA, Tech. Rep., Jan. 2004. DOI: 10.21236/ADA465464. Accessed: Apr. 14, 2026. [Online]. Available: https://apps.dtic.mil/sti/citations/tr/ADA465464.

[2] O. Berthold, H. Federrath, and S. Köpsell, "Web MIXes: A System for Anonymous and Unobservable Internet Access," in *Designing Privacy Enhancing Technologies*, G. Goos, J. Hartmanis, J. Van Leeuwen, and H. Federrath, Eds., vol. 2009, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 115–129, ISBN: 978-3-540-41724-8 978-3-540-44702-3. DOI: 10.1007/3-540-44702-4_7. Accessed: Apr. 14, 2026. [Online]. Available: http://link.springer.com/10.1007/3-540-44702-4_7.

[3] *JAP – ANONYMITY & PRIVACY*. Accessed: Oct. 14, 2025. [Online]. Available: https://anon.inf.tu-dresden.de/.

[4] C. Davis, *IPSec: Securing VPNs* (Rsa Press S). Osborne/McGraw-Hill, 2001, ISBN: 978-0-07-212757-7. [Online]. Available: https://books.google.com/books?id=VqGwNQAACAAJ.

[5] A. Hintz, "Fingerprinting websites using traffic analysis," in *Proceedings of the 2nd international conference on Privacy enhancing technologies*, ser. PET'02, Berlin, Heidelberg: Springer-Verlag, Apr. 2002, pp. 171–178, ISBN: 978-3-540-00565-0. Accessed: Apr. 14, 2026.

[6] M. Liberatore and B. N. Levine, "Inferring the source of encrypted HTTP connections," en, in *Proceedings of the 13th ACM conference on Computer and communications security*, Alexandria Virginia USA: ACM, Oct. 2006, pp. 255–263, ISBN: 978-1-59593-518-2. DOI: 10.1145/1180405.1180437. Accessed: Apr. 14, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/1180405.1180437.

[7] G. D. Bissias, M. Liberatore, D. Jensen, and B. N. Levine, "Privacy Vulnerabilities in Encrypted HTTP Streams," in *Privacy Enhancing Technologies*, G. Danezis and D. Martin, Eds., vol. 3856, Series Title: Lecture Notes in Computer Science, Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 1–11, ISBN: 978-3-540-34745-3 978-3-540-34746-0. DOI: 10.1007/11767831_1. Accessed: Apr. 14, 2026. [Online]. Available: http://link.springer.com/10.1007/11767831_1.

[8] B. Yee, *Personal communication*, Cited in Wagner Schneir's Analysis of the SSL 3.0 Protocol, 1996.

[9] D. Wagner and B. Schneier, "Analysis of the {SSL} 3.0 Protocol," en, 1996. Accessed: Apr. 21, 2026. [Online]. Available: https://www.usenix.org/conference/2nd-usenix-workshop-electronic-commerce/analysis-ssl-30-protocol.

[10] H. Cheng, "Traffic Analysis of SSL Encrypted Web Browsing," 1998. Accessed: Apr. 21, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Traffic-Analysis-of-SSL-Encrypted-Web-Browsing-Cheng/1a987c4fe65fa347a863dece665955ee7e01791b.

[11] Qixiang Sun, D. Simon, Yi-Min Wang, W. Russell, V. Padmanabhan, and Lili Qiu, "Statistical identification of encrypted Web browsing traffic," in *Proceedings 2002 IEEE Symposium on Security and Privacy*, Berkeley, CA, USA: IEEE Comput. Soc, 2002, pp. 19–30, ISBN: 978-0-7695-1543-4. DOI: 10.1109/SECPRI.2002.1004359. Accessed: Apr. 16, 2026. [Online]. Available: http://ieeexplore.ieee.org/document/1004359/.

[12] D. Herrmann, R. Wendolsky, and H. Federrath, "Website fingerprinting: Attacking popular privacy enhancing technologies with the multinomial naïve-bayes classifier," en, in *Proceedings of the 2009 ACM workshop on Cloud computing security*, Chicago Illinois USA: ACM, Nov. 2009, pp. 31–42, ISBN: 978-1-60558-784-4. DOI: 10.1145/1655008.1655013. Accessed: Apr. 14, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/1655008.1655013.

[13] A. Al Azzani and A. Shaawa, "Evaluating the Effectiveness of Anonymization Systems Against Website Fingerprinting Attacks Using Machine Learning Algorithms: Implementation and Countermeasures," in *2024 1st International Conference on Emerging Technologies for Dependable Internet of Things (ICETI)*, Sana'a, Yemen: IEEE, Nov. 2024, pp. 1–5, ISBN: 979-8-3315-3355-7. DOI: 10.1109/ICETI63946.2024.10777244. Accessed: Aug. 4, 2025. [Online]. Available: https://ieeexplore.ieee.org/document/10777244/.

[14] T. Wang, X. Cai, R. Nithyanand, R. Johnson, and I. Goldberg, "Effective Attacks and Provable Defenses for Website Fingerprinting," en, 2014, pp. 143–157, ISBN: 978-1-931971-15-7. Accessed: Apr. 14, 2026. [Online]. Available: https://www.usenix.org/conference/usenixsecurity14/technical-sessions/presentation/wang_tao.

[15] A. Panchenko, L. Niessen, A. Zinnen, and T. Engel, "Website fingerprinting in onion routing based anonymization networks," en, in *Proceedings of the 10th annual ACM workshop on Privacy in the electronic society*, Chicago Illinois USA: ACM, Oct. 2011, pp. 103–114, ISBN: 978-1-4503-1002-4. DOI: 10.1145/2046556.2046570. Accessed: Apr. 14, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/2046556.2046570.

[16] K. He, X. Zhang, S. Ren, and J. Sun, *Deep Residual Learning for Image Recognition*, arXiv:1512.03385 [cs.CV], Dec. 2015. DOI: 10.48550/arXiv.1512.03385. Accessed: May 28, 2026. [Online]. Available: http://arxiv.org/abs/1512.03385.

[17] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, *Densely Connected Convolutional Networks*, arXiv:1608.06993 [cs.CV], Jan. 2018. DOI: 10.48550/arXiv.1608.06993. Accessed: May 28, 2026. [Online]. Available: http://arxiv.org/abs/1608.06993.

[18] K. Abe and S. Goto, "Fingerprinting Attack on Tor Anonymity using Deep Learning," Aug. 2016. Accessed: Apr. 21, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Fingerprinting-Attack-on-Tor-Anonymity-using-Deep-Abe-Goto/bea29c56105b1ef4c4b907a3ffea78cd297cd1c1.

[19] V. Rimmer, D. Preuveneers, M. Juarez, T. V. Goethem, and W. Joosen, "Automated Website Fingerprinting through Deep Learning," en, in *Proceedings 2018 Network and Distributed System Security Symposium*, San Diego, CA: Internet Society, 2018, ISBN: 978-1-891562-49-5. DOI: 10.14722/ndss.2018.23105. Accessed: Apr. 14, 2026. [Online]. Available: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-1_Rimmer_paper.pdf.

[20] P. Sirinam, M. Imani, M. Juarez, and M. Wright, "Deep Fingerprinting: Undermining Website Fingerprinting Defenses with Deep Learning," en, in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, Toronto Canada: ACM, Oct. 2018, pp. 1928–1943, ISBN: 978-1-4503-5693-0. DOI: 10.1145/3243734.3243768. Accessed: Apr. 14, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/3243734.3243768.

[21] S. Bhat, D. Lu, A. Kwon, and S. Devadas, "Var-CNN: A Data-Efficient Website Fingerprinting Attack Based on Deep Learning," en, *Proc. on Priv. Enhancing Technol.*, vol. 2019, no. 4, pp. 292–310, Oct. 2019, ISSN: 2299-0984. DOI: 10.2478/popets-2019-0070. Accessed: Apr. 14, 2026. [Online]. Available: https://petsymposium.org/popets/2019/popets-2019-0070.php.

[22] P. Sirinam, N. Mathews, M. S. Rahman, and M. Wright, "Triplet Fingerprinting: More Practical and Portable Website Fingerprinting with N-shot Learning," en, in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, London United Kingdom: ACM, Nov. 2019, pp. 1131–1148, ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3354217. Accessed: Aug. 6, 2025. [Online]. Available: https://dl.acm.org/doi/10.1145/3319535.3354217.

[23] S. Oh, N. Mathews, M. S. Rahman, M. Wright, and N. Hopper, *GANDaLF: GAN for Data-Limited Fingerprinting*. Dec. 2020, vol. 2021, Journal Abbreviation: Proceedings on Privacy Enhancing Technologies Publication Title: Proceedings on Privacy Enhancing Technologies. DOI: 10.2478/popets-2021-0029.

[24] J. Lu et al., "GAP-WF: Graph Attention Pooling Network for Fine-grained SSL/TLS Website Fingerprinting," in *2021 International Joint Conference on Neural Networks (IJCNN)*, ISSN: 2161-4407, Jul. 2021, pp. 1–8. DOI: 10.1109/IJCNN52387.2021.9533543. Accessed: Apr. 21, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/9533543.

[25] Y. Yuan, W. Zou, and G. Cheng, "Mw3f: Improved multi-tab website fingerprinting attacks with transformer-based feature fusion," *J. Netw. Comput. Appl.*, vol. 236, p. 104 125, 2025, ISSN: 1084-8045. DOI: https://doi.org/10.1016/j.jnca.2025.104125. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1084804525000220.

[26] A. Shaawa and A. A. Azzani, "A Comprehensive Analysis of Website Fingerprinting on the Java Anon Proxy (JAP) using Machine and Deep Learning," *IEEE Access*, pp. 1–1, 2026, ISSN: 2169-3536. DOI: 10.1109/ACCESS.2026.3685551. Accessed: Apr. 24, 2026. [Online]. Available: https://ieeexplore.ieee.org/document/11488195/.

[27] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," en, *Commun. ACM*, vol. 24, no. 2, pp. 84–90, Feb. 1981, ISSN: 0001-0782, 1557-7317. DOI: 10.1145/358549.358563. Accessed: May 30, 2026. [Online]. Available: https://dl.acm.org/doi/10.1145/358549.358563.

[28] Y. Cui et al., *A comprehensive survey of website fingerprinting attacks and defenses in tor: Advances and open challenges*, 2026. arXiv: 2510.11804 [cs.CR]. [Online]. Available: https://arxiv.org/abs/2510.11804.