Evaluating White-Box Adversarial Attacks on Encrypted DGA and DNS Tunneling Detection

Sergio Elizalde*, Ali AlSabeh*, Ali Mazloum*, Jaime Galán-Jiménez†, Elie Kfoury*, Jorge Crichigno*
*University of South Carolina - United States of America
†University of Extremadura - Spain
Emails: elizalds@email.sc.edu, ali.alsabeh@usca.edu, amazloum@email.sc.edu, jaime@unex.es,
ekfoury@email.sc.edu, jcrichigno@cec.sc.edu

***Abstract*—Domain Generation Algorithms (DGAs) and DNS tunneling are widely used to conceal command-and-control and data exfiltration within DNS traffic. The adoption of encrypted DNS protocols, such as DNS over HTTPS (DoH), further complicates detection by removing access to payload contents and forcing reliance on traffic metadata. Machine learning (ML) is commonly used to detect encrypted DNS abuse, but learning-based detectors introduce new attack surfaces. Adversaries can exploit targeted evasion by introducing small, structured perturbations that preserve malicious behavior while inducing misclassification. However, most adversarial ML research focuses on image domains and largely overlooks the inter-feature dependencies inherent to network traffic. In this work, we analyze targeted adversarial evasion against encrypted DNS detectors in a four-class setting comprising DoH-benign, non-DoH-benign (e.g., Web traffic), DGA, and DNS tunneling traffic. We generate constraint-aware adversarial examples using gradient-based attacks, including Projected Gradient Descent (PGD) and Fast Gradient Sign Method (FGSM), on a surrogate neural network, and enforce semantic consistency through a variational autoencoder (VAE). We further evaluate attack transferability to a Random Forest classifier in a gray-box setting. Our results show that PGD achieves up to 80% transfer success and that VAE-guided attacks incur fewer violations of inter-feature constraints than vanilla feature-space perturbations.**

***Index Terms*—Adversarial machine learning, targeted evasion, DoH, Domain Generation Algorithms (DGA), DNS tunneling.**

I. Introduction

Domain Generation Algorithms (DGAs) and DNS tunneling have become major threats in modern networks. They allow malware to hide command-and-control (C2) communication and data exfiltration inside DNS traffic, making detection difficult. This challenge is further amplified when DNS is encrypted. The adoption of DNS over HTTPS (DoH) removes direct access to DNS payloads, forcing defenders to rely on traffic statistics to detect anomaly [1]. Attackers exploit this reduced visibility to blend malicious behavior into normal encrypted DNS traffic, allowing DGA-based communication and tunneling to bypass traditional monitoring tools [2], [3].

Machine Learning (ML) is widely used to detect DGA activity and DNS tunneling in encrypted settings by learning patterns from lexical, temporal, and flow-level features [4]. These approaches have shown strong detection performance and are increasingly deployed in operational environments. However, as ML becomes a core component of Network Intrusion Detection System (NIDS), it also becomes part of the attack surface, introducing new risks [5].

One of the main risks is adversarial evasion. An attacker can apply small, targeted changes to input features to cause a model to misclassify malicious traffic as benign, while preserving the underlying attack behavior. Although adversarial machine learning has been widely studied, most existing work focuses on image-based domains and does not account for the constraints of network traffic. In network data, features are often interdependent and constrained by protocol behavior, timing, and aggregation rules. Ignoring these relationships can result in adversarial samples that are unrealistic or easy to detect in real-world deployments [6].

In this work, we study adversarial evasion attacks against ML-based detectors of encrypted DNS abuse. The adversarial objective is limited to targeted evasion, in which malicious DGA-based communication and DNS tunneling traffic is intentionally perturbed to be misclassified as benign. Our threat model assumes an attacker with full knowledge of the system (i.e., a white-box setting) who generates adversarial samples against a neural network classifier using the Adversarial Robustness Toolbox (ART) [7]. To ensure practical relevance, we apply feature-level constraints and preserve key inter-feature relationships inherent to encrypted DNS traffic via a Variational Autoencoder (VAE). We evaluate the effectiveness of adversarial generation and analyze transferability to a Random Forest (RF) model, widely adopted in network intrusion detection [8]. The main contributions of this paper are:

- A publicly available dataset that augments existing encrypted DNS traffic with realistic DGA-based malware behavior over DoH [9].
- An adversarial threat modeling methodology tailored to evasion of ML-based detection in encrypted DNS traffic.
- A practical adversarial generation pipeline that preserves inter-feature dependencies by using VAE.
- A comparative evaluation of gradient-based techniques for generating adversarial examples targeting DGA and DNS tunneling detection.

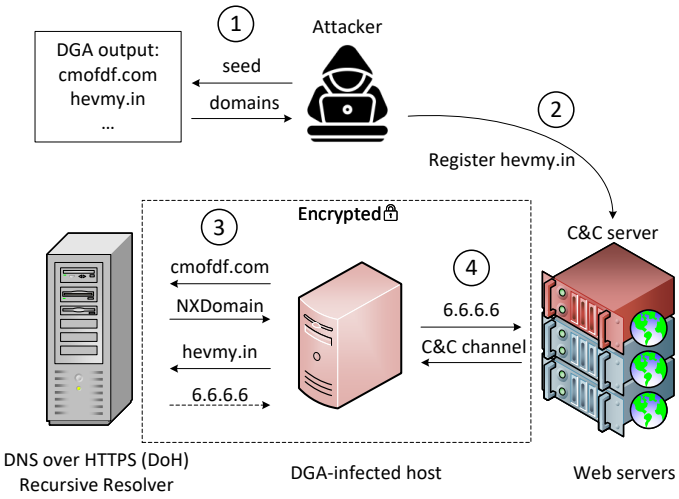

Fig. 1: Illustrative workflow of a DGA–based command-and-control (C2) attack over encrypted DNS using DNS over HTTPS.

II. Background

A. Encrypted DNS & DGA threats

DGAs are widely used by malware to conceal C2 infrastructure by dynamically generating large sets of pseudo-random domain names. When combined with DoH, this activity is further obfuscated, as DNS queries are encrypted and payload-level visibility is eliminated.

As illustrated in Fig. 1, an attacker initializes DGA using a seed (e.g., timestamp) to generate candidate domains and registers a selected domain that resolves to a C2 server. A compromised host executes the same DGA and issues DNS queries for these domains over DoH. Because the queries are encapsulated within encrypted HTTPS traffic, in-network defenses cannot directly inspect domain names. The DoH resolver decrypts the queries and attempts resolution through the DNS hierarchy. This process may generate multiple NXDOMAIN responses before reaching a registered domain; however, these intermediate NXDOMAINs are confined to the resolver and are not visible on the network, as all client–resolver communication remains encrypted. Once resolved, the resolver returns the C2 server's IP address, enabling command-and-control communication.

While DNS payloads remain encrypted, flow-level metadata associated with DoH traffic remains observable. This residual visibility underpins the adversarial feature analysis and detection methodology proposed in this work.

B. Adversarial Machine Learning

Adversarial Machine Learning (AML) studies how malicious actors manipulate machine learning systems to evade detection or degrade performance. Recent NIST guidance highlights that AML threats can arise across the entire ML lifecycle, from data collection and training to deployment and inference [10]. This work focuses on *evasion attacks*, which target the inference phase by crafting malicious inputs that induce misclassification without modifying the training process. Such attacks are particularly relevant in operational NIDS deployments, as they can be executed at runtime.

We consider both white-box and gray-box threat models. In the white-box setting, the attacker has full knowledge of the model architecture, parameters, and feature representation, enabling gradient-based evasion. In more realistic gray-box settings, the attacker has only partial knowledge, such as access to similar traffic data or feature definitions, and relies on attack transferability. Due to the structured and interdependent nature of network traffic features, evasion attacks against NIDS are constrained by semantic validity, limiting the direct applicability of adversarial techniques developed for image-based domains [11].

III. Related Work

A. Encrypted DNS Tunneling and DGA detection

Prior studies have investigated DNS abuse, including covert communication channels and DGAs, in both plaintext and encrypted DNS settings. Alsabeh et al. [2] proposed a P4-based DGA detection framework using bigram frequency features and a random forest classifier, primarily targeting plaintext DNS. The scheme uses P4 switches [12] to operate at line rate. Other works focus on DGA detection in encrypted DNS, employing machine learning models such as multilayer perceptrons and hierarchical tree-based ensembles for real-time or multi-class classification [4], [13]. In contrast, Moure-Garrido et al. [14] introduced a heuristic-based approach for encrypted DNS flow analysis, achieving strong offline performance but relying on fixed set of empirically derived features that may limit scalability as threat behaviors evolve. These approaches overlook the models' inherent vulnerabilities to adversarial attacks.

B. Adversarial ML for network intrusion detection

Research in adversarial machine learning has shown that ML models used in cybersecurity are inherently vulnerable to carefully crafted perturbations that induce misclassifications and reduce confidence scores [15]. Prior research has systematized adversarial threats against machine learning systems by categorizing attacks on learning components, particularly in malware and command-and-control (C2) detection, into evasion and poisoning attacks, revealing that many models were not designed for adversarial settings [16]. This line of work was extended through structured threat modeling approaches, such as attack trees, to reason about the feasibility and impact of adversarial attacks across confidentiality, integrity, and availability objectives [17]. Complementary empirical evaluations in networked and IoT environments have shown that state-of-the-art models can experience substantial performance degradation under white-box, gray-box, and black-box attacks, including FGSM and PGD [18].

More recent studies in network security domains have shown that many adversarial samples generated by unconstrained white-box attacks violate protocol semantics or numerical feasibility, limiting their applicability in real deploy-

TABLE I: Bidirectional flow features.

Features		Stats	Scope		
			Client	Server	Global
Size	IPv4 Packet	Min	●	●	●
		Max	●	●	●
		Avg.	●	●	●
Context	Dir. switches	Count	○	○	●
	Packets	Count	●	●	○
	Packet fraction	Ratio	●	○	○
	Bytes	Count	●	●	○
	Bytes fraction	Ratio	●	○	○

● Selected ○ Not used

ments [19]. Motivated by these findings, this paper advances the state of the art by examining adversarial example generation in flow-based encrypted DNS detection, where feasibility constraints arise naturally from packet aggregation, directionality, and traffic statistics. We investigate how gradient-based attacks behave under these constraints and evaluate the impact of constraint-aware generation on both evasion effectiveness and gray-box transferability.

IV. Threat Model

A. Adversarial setting

Attacker goal. The adversary generates malicious encrypted DoH traffic (e.g., DGA-based C2 and/or DNS tunneling) while evading an ML-based traffic classifier. We consider *targeted evasion at inference time*: a malicious flow is perturbed so that the detector predicts a chosen benign label (e.g., DoH-benign or non-DoH benign). We quantify success using *attack success rate (ASR)*, i.e., the fraction of malicious samples mapped to the intended benign target class.

Attacker knowledge. We assume a *white-box* adversary with knowledge of the dataset used, feature extraction pipeline, normalization, and the parameters of the deployed MLP detector, enabling full visibility to launch gradient-based attacks. To approximate realistic deployments, we also evaluate a *gray-box* setting via transferability: adversarial samples crafted on a surrogate differentiable model are tested against a non-differentiable model (e.g., Random Forest) trained on the same features.

Attacker control. The attacker controls an infected host and can shape observable DoH traffic statistics (within protocol constraints), e.g., by adjusting request rates, batching, padding, fragmentation, and other host-level behaviors that affect flow-level features extracted from network traffic. The attacker does *not* control DoH resolvers/servers and cannot tamper with the monitoring infrastructure.

B. Flow-Level Features and Feasibility Constraints

Previous work on encrypted traffic classification has shown that, despite payload encryption, statistical patterns extracted from packet metadata remain effective for identifying DoH traffic [20]. Building on these findings, this work focuses on lightweight, bidirectional flow-level features that can be computed efficiently.

Feature collection window. Each bidirectional flow is represented using statistical summaries derived from the first N packets, where N defines the feature collection window. In this study, N was empirically set to 16, as it represents the smallest window that achieves stable classification performance. Increasing N beyond this value yielded negligible accuracy improvements while introducing additional latency and processing overhead, which are critical for timely response.

Feature set. Table I summarizes the final set of features used by the proposed classifier. The selected features fall into two main categories: (i) *size-based features*, which characterize packet size distributions using minimum, mean, and maximum packet sizes, and (ii) *contextual features*, which capture flow behavior such as packet counts, byte counts, direction switches (i.e., packet source direction alternations), and client-side packet and byte fractions. Features are further defined by scope, indicating whether they are computed over client-side packets, server-side packets, or the entire bidirectional flow.

Feasibility constraints. All features are computed from the same fixed-length window of N packets from a *bidirectional* DoH exchange (client $\rightarrow$ server and server $\rightarrow$ client). Because these features summarize the *same* packet sequence, they are not independent.

Notation. Let n_c and n_s be the number of packets sent by the client and server within the window, respectively. Let B_c and B_s be the corresponding total bytes, and $B = B_c + B_s$ the total bytes in the N packets. We define the *packet share* and *byte share* of the client as

$$p_c \triangleq \frac{n_c}{N}, \qquad b_c \triangleq \frac{B_c}{B},$$

with server shares $p_s = 1 - p_c$ and $b_s = 1 - b_c$. Let $\ell_{\min}$, $\bar{\ell}$, and $\ell_{\max}$ denote the minimum, mean, and maximum packet size (in bytes) over the N packets (and similarly per-direction when those statistics are used). Finally, let S be the number of direction changes in the packet sequence.

A feature vector is feasible only if it satisfies:

- *Packet counts add up:*

$$n_c + n_s = N, \qquad n_c, n_s \geq 0.$$

- *Fractions are valid:*

$$0 \leq p_c \leq 1, \quad 0 \leq b_c \leq 1,$$

- *Size statistics must be ordered:*

$$\ell_{\min} \leq \bar{\ell} \leq \ell_{\max},$$

- *Byte totals must match counts and means:*

$$B = N\bar{\ell}, \qquad B_c = n_c\bar{\ell}_c, \qquad B_s = n_s\bar{\ell}_s$$

  whenever $\bar{\ell}_c$ and $\bar{\ell}_s$ are included.
- *Direction switches are bounded:*

$$0 \leq S \leq N - 1, \qquad S \leq 2\min(n_c, n_s).$$

These relationships define the feasible region of flow-level features for an N-packet DoH exchange. Adversarial perturbations must conserve these constraints to be valid.

V. Adversarial Attacks

We evaluate the robustness of the proposed classifier against *targeted evasion attacks* using two gradient-based methods: FGSM and PGD. Both attacks are incorporated in our pipeline using the ART Toolbox from IBM [7]. The adversary controls malicious traffic and aims to modify its feature representation such that it is misclassified as a specific benign class (e.g., benign DoH or non-DoH). Accordingly, all attacks are targeted and are applied exclusively to malicious samples.

A. Fast Gradient Sign Method (FGSM)

FGSM is a single-step attack that computes an adversarial example $\mathbf{x}'$ by perturbing the input $\mathbf{x}$ in the direction that minimizes the loss for a chosen target class y_t:

$$\mathbf{x}' = \mathbf{x} - \epsilon \cdot \text{sign}(\nabla_{\mathbf{x}} \mathcal{L}(\mathbf{x}, y_t)) \,,$$

where ϵ bounds the maximum per-feature perturbation. The sign operator ensures that each feature is modified by at most ϵ, yielding the largest change in loss under an L_∞ constraint in a single step.

B. Projected Gradient Descent (PGD)

PGD extends FGSM by applying multiple smaller perturbation steps while enforcing the same L_∞ constraint. Starting from the original input, PGD iteratively updates:

$$\mathbf{x}_{k+1} = \text{Proj}_{\|\cdot - \mathbf{x}\|_\infty \leq \epsilon}(\mathbf{x}_k - \alpha \cdot \text{sign}(\nabla_{\mathbf{x}_k} \mathcal{L}(\mathbf{x}_k, y_t))) \,,$$

where α is the step size. The projection operator ensures that the accumulated perturbation remains within the allowed budget relative to the original input.

C. Feature-constrained attack

Fig. 2 illustrates the feature-constrained adversarial attack pipeline under a surrogate white-box threat model. Instead of directly perturbing flow-level features, we constrain adversarial optimization using a VAE trained on the standardized training data. The VAE consists of a tabular encoder and decoder with two hidden layers (256 units each) and a 16-dimensional latent space.

During training, the encoder learns to map each flow to a latent distribution characterized by a mean and variance, while the decoder reconstructs the original feature vector. The model is optimized to minimize reconstruction error while regularizing the latent space to follow a smooth Gaussian structure. At attack time, we use the encoder's latent mean as a deterministic representation of each flow.

Targeted FGSM and PGD attacks are then applied directly in this latent space by optimizing the classifier's loss after decoding, i.e., latent vectors are perturbed so that their decoded features are misclassified as a chosen benign class. The perturbed latent vectors are decoded back into feature space, producing adversarial samples that remain close to the learned manifold of valid network flows. The encoder and decoder are frozen during attack generation.

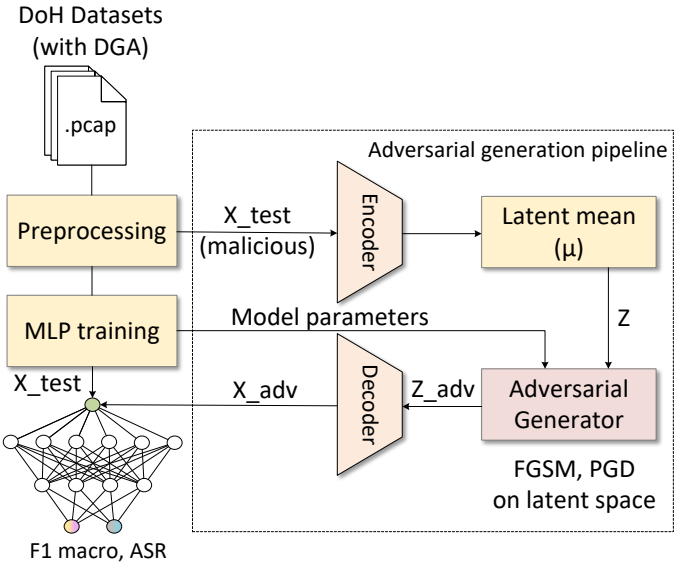

Fig. 2: Latent-space adversarial attack pipeline using a VAE.

VI. Experimental Evaluation

A. Datasets and processing

This study relies on publicly available datasets that capture both benign and malicious DNS over HTTPS (DoH) traffic, including Domain Generation Algorithm (DGA) activity and DNS tunneling behavior. The datasets used in this study are summarized as follows:

- **CIRA-CIC-DoHBrw-2020** [21]: Contains benign HTTPS and DoH browsing traffic, along with DNS tunneling flows generated using tools such as *iodine*, *dns2tcp*, and *dnscat2*.
- **DoH-DGA-Malware-Traffic-HKD** [13]: Consists of DoH traffic produced by real malware families (e.g., *PadCrypt*, *Sisron*, *Tinba*, *Zloader*), capturing realistic DGA-based command-and-control behavior.
- **Augmented DoH dataset** [9]: Expands the available DGA samples by replaying plaintext DNS PCAPs as DoH traffic using real DoH resolvers (e.g., Cloudflare, Google, Quad9) over HTTPS, and combining the resulting encrypted flows with malware-generated DoH traces.

All PCAPs were processed using the `dpkt` library to extract bidirectional flow-level statistics. For each flow, we computed the same set of 16 metadata features over a fixed window of $N = 16$ packets, capturing packet size, directionality, and byte distribution characteristics. Flows originating from different datasets but representing the same class were grouped under a common label, yielding four classes: `DoH_Benign`, `NonDoH_Benign`, `DGA`, and `DoH_Tunnel`. To obtain a balanced evaluation corpus, each class was downsampled using a fixed random seed (`42`), resulting in a balanced dataset of approximately **200,000** flows per class. Only numeric features were retained and standardized using a `StandardScaler`

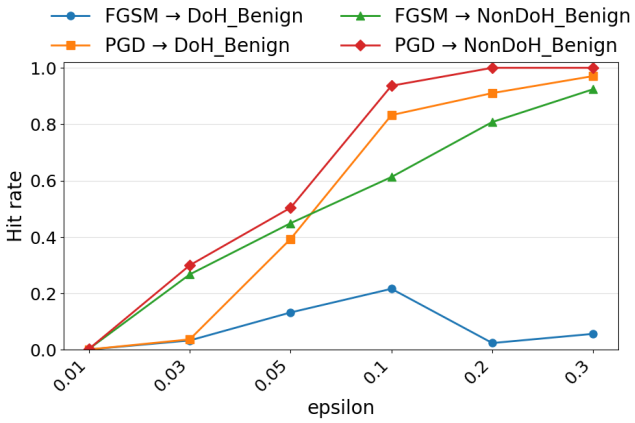

Fig. 3: Vanilla attack success rate using FGSM and PGD.

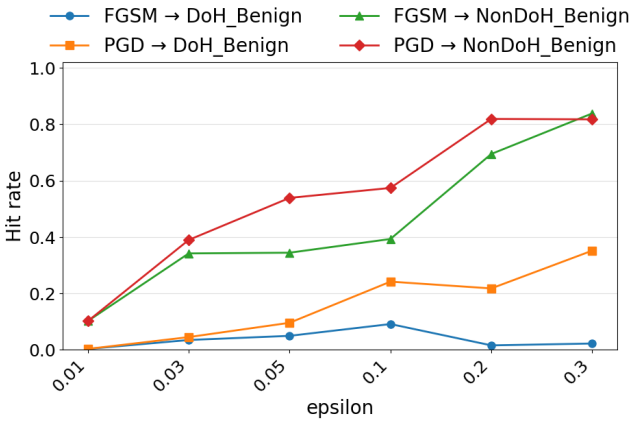

Fig. 4: Transferability of attacks to RF classifier.

fit on the training data. The final dataset was split into training and test sets using a 70%:30% partition.

B. Metric

We evaluate adversarial evasion using two metrics: *Attack Success Rate (ASR)* and *violation count*. ASR quantifies the effectiveness of targeted evasion, while the violation count measures how well adversarial samples comply with the feature-level feasibility constraints defined in Section IV.

1) Attack Success Rate (ASR): Attack Success Rate measures the fraction of malicious samples for which a targeted adversarial attack achieves its intended outcome. In this work, targeted evasion aims to misclassify malicious DoH traffic as a specified benign class.

Formally, let $\mathbf{x}_i$ denote a malicious sample, $\mathbf{x}'_i$ its adversarial counterpart, and y_t the target benign label. ASR is defined as:

$$\text{ASR} = \frac{1}{M} \sum_{i=1}^{M} \mathbb{I}\left[f(\mathbf{x}'_i) = y_t\right], \tag{1}$$

where M is the number of malicious samples and $f(\cdot)$ is the classifier.

C. Adversarial attacks

1) Vanilla attacks using FGSM and PGD: We evaluate baseline targeted evasion using vanilla FGSM and PGD attacks applied directly in feature space, targeting an MLP classifier with two hidden layers of sizes 64 and 32. To maintain feasibility, adversarial features are clipped to the 1^{st} and 99^{th} percentiles of the training data, preventing out-of-range values without enforcing explicit constraints. The perturbation budget ϵ is swept for both attacks, controlling the step size in FGSM and the per-iteration step size in PGD.

The evaluation is performed on malicious DGA and DNS-tunneling test samples, with approximately 17,500 test samples per target class. Since each malicious sample is attacked toward both `DoH_Benign` and `NonDoH_Benign`, this results in 35,000 adversarial samples generated.

Fig. 3 illustrates the targeted attack success rate as a function of the perturbation budget. As ϵ increases, PGD consistently outperforms FGSM, exhibiting a largely monotonic increase in attack success for both benign targets. In contrast, FGSM shows unstable and non-monotonic behavior, particularly when targeting `DoH_Benign`, where increasing perturbation budgets do not reliably translate into higher success. This suggests that FGSM perturbations can move samples away from the effective decision boundary rather than toward the target region. The results indicate that PGD provides a more reliable and controllable evasion mechanism and that the `NonDoH_Benign` class is inherently more accessible from malicious traffic than `DoH_Benign`.

2) Gray-box Transferability to RF: To assess attack effectiveness beyond the white-box setting, we evaluate the transferability of adversarial examples in a gray-box scenario. Adversarial samples are generated using the MLP surrogate model and directly applied to a random forest (RF) classifier trained on the same feature set. The RF consists of five estimators and represents a non-differentiable model commonly used for efficient online deployment.

Fig. 4 summarizes targeted attack transferability to the RF classifier. In the gray-box setting, PGD consistently outperforms FGSM, but overall attack effectiveness is reduced compared to the white-box MLP case, with maximum transfer success capped at around 80%. Transferability remains substantially higher when targeting `NonDoH_Benign`, whereas attacks against `DoH_Benign` are significantly less effective, saturating at approximately 30–35% even under PGD. This reduction highlights the increased robustness of the RF classifier and reaffirms that `DoH_Benign` is inherently more challenging to reach from malicious traffic across model architectures.

3) Latent-Space Targeted Attacks: Table II reports results for latent-space targeted attacks using a fixed perturbation budget of $\epsilon = 0.3$. Attacks are applied to malicious test samples only and target benign classes. The VAE is trained on the training split and configured with a 16-dimensional latent space and encoder/decoder networks with two 256-unit hidden layers. During attack generation, malicious samples are encoded using the latent mean, perturbed using FGSM or PGD

TABLE II: Vanilla vs VAE guided adversarial attack

Target	Method	Pipeline	ASR %	Avg. constraints violations
DoH	PGD	Vanilla	92.3	5.01
		VAE	**52.5**	**3.43**
	FGSM	Vanilla	39	4.62
		VAE	41	3.95
Non-DoH	PGD	Vanilla	99.9	5.07
		VAE	20.7	3.72
	FGSM	Vanilla	18	4.88
		VAE	20	4.01

in latent space, and decoded back into feature space.

In terms of ASR, VAE-guided attacks remain effective and exhibit the same relative behavior observed in earlier experiments. When targeting `DoH_Benign`, VAE-guided PGD achieves attack success rates above 50%, while attacks against `NonDoH_Benign` remain easier overall, reflecting a looser decision boundary for non-DoH traffic. This consistency indicates that constraining adversarial generation to the latent space preserves the relative feasibility of evasion across target classes while enforcing stronger structural constraints on the generated samples.

To assess realism, we count violations of the feasibility constraints defined in Section IV. No adversarial sample fully satisfies all constraints, with the most frequent violation being the packet-count consistency requirement ($n_c + n_s = 16$). However, VAE-guided attacks consistently reduce the average number of violations compared to vanilla attacks, particularly by preserving ordering relationships among minimum, mean, and maximum size statistics.

VII. Conclusion and Future Work

This paper evaluates gradient-based targeted evasion against ML detectors of encrypted DNS abuse and draws three practical insights. First, even lightweight classifiers commonly used in network settings (an MLP surrogate and an RF baseline) remain vulnerable: iterative PGD consistently outperforms FGSM, and relatively modest, semi-constrained perturbations achieve high targeted success rates. Second, adversarial examples exhibit substantial *transferability*, with samples crafted on a white-box MLP degrading RF performance by up to roughly 80%, indicating that breaking a differentiable surrogate can meaningfully impact non-differentiable deployments. Third, constraining adversarial generation to a learned manifold via a VAE yields adversarial samples that better preserve inter-feature dependencies and thus appear more realistic than vanilla feature-space perturbations. Future work should explore: (i) translating feature perturbations into concrete packet-level manipulations to assess end-to-end feasibility, and (ii) developing and evaluating practical defenses capable of detecting or mitigating adversarial manipulations.

Acknowledgment

The work was supported by the National Science Foundation under awards 2417823 and 2403360.

References

[1] M. Shen, Y. Liu, L. Zhu, K. Xu, X. Du, and N. Guizani, "Optimizing feature selection for efficient encrypted traffic classification: A systematic approach," *IEEE Network*, vol. 34, no. 4, pp. 20–27, 2020.

[2] A. AlSabeh, K. Friday, E. Kfoury, J. Crichigno, and E. Bou-Harb, "On dga detection and classification using p4 programmable switches," *Computers & Security*, vol. 145, p. 104007, 2024.

[3] A. AlSabeh, K. Friday, J. Crichigno, and E. Bou-Harb, "Effective dga family classification using a hybrid shallow and deep packet inspection technique on p4 programmable switches," in *IEEE International Conference on Communications*, 2023.

[4] A. R. Tapsoba, T. F. Ouédraogo, M. B. Diallo, and W.-B. S. Zongo, "Toward real time dga domains detection in encrypted traffic," in *Proceedings of the 7th International Conference on Networking, Intelligent Systems and Security*, 2024.

[5] A. Paleyes, R.-G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: a survey of case studies," *ACM computing surveys*, vol. 55, no. 6, pp. 1–29, 2022.

[6] S. Ennaji, F. De Gaspari, D. Hitaj, A. Kbidi, and L. V. Mancini, "Adversarial challenges in network intrusion detection systems: Research insights and future prospects," *IEEE Access*, vol. 13, pp. 148 613–148 645, 2025.

[7] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.2.0," *CoRR*, vol. 1807.01069, 2018. [Online]. Available: https://arxiv.org/pdf/1807.01069

[8] A. Mazloum, A. AlSabeh, E. Kfoury, and J. Crichigno, "Domain name security inspection at line rate: Tls sni extraction in the data plane using p4 and dpdk," in *IEEE International Conference on Communications*, 2025.

[9] CI lab, "DGA dataset," [Online]. Available: https://tinyurl.com/3s7z4zpw, accessed: 2025-11-10.

[10] A. Vassilev, A. Oprea, A. Fordyce, and H. Andersen, "Adversarial machine learning: A taxonomy and terminology of attacks and mitigations," 2024.

[11] K. He, D. D. Kim, and M. R. Asghar, "Adversarial machine learning for network intrusion detection systems: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 1, pp. 538–566, 2023.

[12] E. Kfoury, J. Crichigno, and E. Bou-Harb, "An exhaustive survey on p4 programmable data plane switches: Taxonomy, applications, challenges, and future trends," *IEEE Access*, vol. 9, 2021.

[13] R. Mitsuhashi, Y. Jin, K. Iida, T. Shinagawa, and Y. Takai, "Detection of dga-based malware communications from doh traffic using machine learning analysis," in *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, 2023.

[14] M. Moure-Garrido, S. K. Das, C. Campo, and C. Garcia-Rubio, "Real-time analysis of encrypted dns traffic for threat detection," in *IEEE International Conference on Communications*, 2024.

[15] K. Roshan and A. Zafar, "Black-box adversarial transferability: An empirical study in cybersecurity perspective," *Computers & Security*, vol. 141, p. 103853, 2024.

[16] J. Gardiner and S. Nagaraja, "On the security of machine learning in malware c&c detection: A survey," *ACM Computing Surveys (CSUR)*, vol. 49, no. 3, pp. 1–39, 2016.

[17] S. V. Hoseini, J. Suutala, J. Partala, and K. Halunen, "Threat modeling ai/ml with the attack tree," *IEEE Access*, vol. 12, pp. 172 610–172 637, 2024.

[18] O. Gungor, E. Li, Z. Shang, Y. Guo, J. Chen, J. Davis, and T. Rosing, "Rigorous evaluation of machine learning-based intrusion detection against adversarial attacks," in *2024 IEEE International Conference on Cyber Security and Resilience (CSR)*. IEEE, 2024, pp. 152–158.

[19] A. Grini, O. Taheri, B. El Khamlichi, and A. El Fallah-Seghrouchni, "Constrained network adversarial attacks: Validity, robustness, and transferability," in *IEEE DCOSS-IoT Conference*, 2025.

[20] M. Lyu, H. H. Gharakheili, and V. Sivaraman, "A survey on dns encryption: Current development, malware misuse, and inference techniques," *ACM Computing Surveys*, vol. 55, no. 8, pp. 1–28, 2022.

[21] M. MontazeriShatoori, L. Davidson, G. Kaur, and A. H. Lashkari, "Detection of doh tunnels using time-series classification of encrypted traffic," in *IEEE DASC/PiCom/CBDCom/CyberSciTech Conference*, 2020.