Computer Standards & Interfaces 97 (2026) 104111

Contents lists available at ScienceDirect

Computer Standards & Interfaces

journal homepage: www.elsevier.com/locate/csi

Refining decision boundaries via dynamic label adversarial training for robust traffic classification☆

Haoyu Tong [a,c,d], Meixia Miao [b,c,d], Yundong Liu [a,c,d], Xiaoyu Zhang [a,c,d,*], Xiangyang Luo [c,d], Willy Susilo [e]

[a] State Key Laboratory of Integrated Service Networks (ISN), Xidian University, 710121, Xi'an, China
[b] School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, 710121, China
[c] Key Laboratory of Cyberspace Security, Ministry of Education of China, 450001, Zhengzhou, China
[d] Henan Key Laboratory of Cyberspace Situation Awareness, 450001, Zhengzhou, China
[e] School of Computing and Information Technology, University of Wollongong, Wollongong, Australia

ARTICLE INFO

Keywords:
Traffic classification
Adversarial examples
Adversarial training
Label noise

ABSTRACT

Network traffic classification plays a critical role in securing modern communication systems, as it enables the identification of malicious or abnormal patterns within traffic data. With the growing complexity of network environments, deep learning models have emerged as a compelling solution due to their ability to automatically learn discriminative representations from raw traffic. However, these models are highly vulnerable to adversarial examples, which can significantly degrade their performance by introducing imperceptible perturbations. While adversarial training (AT) has emerged as a primary defense, it often suffers from label noise, particularly when hard labels are forcibly assigned to adversarial examples whose true class may be ambiguous. In this work, we first analyze the detrimental effect of label noise on adversarial training, revealing that forcing hard labels onto adversarial examples can cause excessive shifts of the decision boundary away from the adversarial examples, which in turn degrades the model's generalization. Motivated by the theoretical analysis, we propose Dynamic Label Adversarial Training (DLAT), a novel AT framework that mitigates label noise via dynamically mixed soft labels. DLAT interpolates the logits of clean and adversarial examples to estimate the labels of boundary-adjacent examples, which are then used as soft labels for adversarial examples. By adaptively aligning the decision boundary toward the vicinity of adversarial examples, the framework constrains unnecessary boundary shifts and alleviates generalization degradation caused by label noise. Extensive evaluations on network traffic classification benchmarks validate the effectiveness of DLAT in outperforming standard adversarial training and its variants in both robustness and generalization.

1. Introduction

Network traffic classification, which aims to determine the application or service associated with observed traffic packets, flows, or sessions, serves as a fundamental building block in a wide range of networking tasks, including intrusion detection, quality-of-service management, and traffic engineering [1,2]. In the early stages of network management, classification was carried out mainly through port-based identification [3,4] and deep packet inspection (DPI) [5,6]. However, these traditional approaches have become increasingly ineffective due to the widespread use of dynamic port allocation, encrypted communication protocols, and intentional obfuscation techniques [7,8]. As network environments become more complex and security-conscious, there is a growing demand for more intelligent and adaptive classification methods that do not rely on payload visibility or fixed port mappings.

In recent years, deep learning (DL) [9] has become a dominant paradigm for network traffic classification due to its ability to automatically extract the underlying representations from raw or lightly processed traffic data [10–14]. Compared to traditional statistical or machine learning approaches that rely heavily on manual feature engineering, deep neural networks, including convolutional, recurrent, and Transformer-based architectures, can effectively capture spatial and temporal patterns in traffic data, enabling high accuracy even in challenging scenarios such as previously unseen traffic. However,

☆ This article is part of a Special issue entitled: 'Secure AI' published in Computer Standards & Interfaces.

* Corresponding author at: State Key Laboratory of Integrated Service Networks (ISN), Xidian University, 710121, Xi'an, China.
E-mail addresses: haoyutong@stu.xidian.edu.cn (H. Tong), miaofeng415@163.com (M. Miao), yundongliu@stu.xidian.edu.cn (Y. Liu), xiaoyuzhang@xidian.edu.cn (X. Zhang), xiangyangluo@126.com (X. Luo), wsusilo@uow.edu.au (W. Susilo).

https://doi.org/10.1016/j.csi.2025.104111
Received 26 October 2025; Received in revised form 29 November 2025; Accepted 8 December 2025
Available online 13 December 2025

despite their impressive performance, deep learning-based classifiers remain highly susceptible to adversarial examples. These are deliberately crafted inputs with imperceptible perturbations that cause models to misclassify [15,16]. In the context of traffic classification, adversarial perturbations can manipulate flow-level features or packet sequences in ways that evade detection without disrupting the underlying communication protocols. To mitigate this vulnerability, adversarial training has been widely adopted as a defense mechanism by introducing adversarial examples during model training to enhance robustness [17].

While adversarial training is effective in many domains, applying it to traffic classification poses unique challenges. Unlike natural image domains, traffic data distributions typically exhibit higher intrinsic dimensionality and more complex manifold structures. Different application protocols often share significant common subsequences at the byte level, creating naturally entangled features that separate classes through subtle statistical patterns rather than distinct visual characteristics. Furthermore, unlike images where semantic meaning is often locally correlated, traffic features exhibit long-range dependencies across packet sequences, making them particularly sensitive to small, strategically placed perturbations. These characteristics cause even minor perturbations to readily shift traffic samples across class boundaries, leading to significant label noise during training. This issue is further exacerbated by standard adversarial training practices [18], which introduce perturbed examples into the training set while still assigning them the same labels as their clean examples, thereby intensifying the semantic mismatch between the true and assigned labels. Traditional adversarial training typically enforces the original hard label on adversarial examples. While effective to some extent, this rigid supervision introduces significant label noise, especially when adversarial examples cross or approach decision boundaries. Consequently, the decision boundary is pushed away from perturbed examples, often reinforcing the robustness of the class in which the adversarial example is located at the expense of others. This imbalance undermines the overall robustness of the model, particularly in tasks such as traffic classification, where class semantics are inherently ambiguous and sensitive to perturbations.

To address this issue, we propose Dynamic Label Adversarial Training (DLAT), a novel adversarial training framework designed to mitigate the adverse effects of excessive label noise in robust network traffic classification. Rather than rigidly assigning the original hard label to adversarial examples, DLAT constructs soft labels for examples near decision boundaries through a similarity-guided strategy that takes advantage of the model's output distributions. Such soft labels help guide the decision boundary toward the neighborhood of adversarial examples, rather than forcing it away due to overconfident and potentially incorrect supervision. Instead of explicitly approximating the decision boundary using computationally intensive techniques, such as multi-step adversarial attacks with decaying step sizes, DLAT leverages the similarity between the output logits of clean and perturbed inputs to estimate the soft labels of the examples near the decision boundary. Specifically, since the similarity between their output distributions reflects how close the adversarial example lies to the current decision boundary, it serves as a reliable proxy for boundary proximity. Based on this similarity, DLAT interpolates between the model's prediction on the clean and adversarial inputs. When adversarial and clean outputs are closely aligned, the soft label remains closer to the clean prediction; on the contrary, greater divergence triggers a softer supervisory signal that better reflects the model's uncertainty regarding adversarial input. This adaptive labeling mechanism mitigates the semantic distortion introduced by fixed-label training, thus reducing the risk of reinforcing incorrect decision boundaries and improving robustness under label noise. Specifically, since the similarity between the output distributions of clean and adversarial examples serves as an effective proxy for their proximity to the decision boundary, DLAT computes this similarity to guide the interpolation between their corresponding logits. When the adversarial example is far from the boundary, a larger weight is assigned to the clean prediction. In contrast, when it is close to the boundary, more weight is allocated to the adversarial output. This similarity-guided interpolation enables precise estimation of soft labels for boundary-adjacent examples, which in turn facilitates more accurate adjustment of the decision boundary. By avoiding rigid supervision of hard labels, this adaptive labeling mechanism mitigates semantic distortion and helps the model learn more robust decision surfaces under label noise. Our key contributions are outlined as follows:

- We extend the understanding of label noise in adversarial training to the domain of network traffic classification. The compact and entangled distribution of traffic data makes it vulnerable to small perturbations, increasing the likelihood of label inconsistency in adversarial examples. This inconsistency corresponds to a higher degree of label noise, which enforces incorrect alignment and impedes the learning of robust decision boundaries.
- We provide a theoretical characterization of how hard-label supervision on shifted adversarial examples induces excessive movement of the decision boundary. Specifically, enforcing high-confidence predictions for adversarial examples distorts the classifier, increasing the risk of misclassification for nearby examples from other classes.
- We introduce a novel adversarial training method called DLAT, which dynamically assigns soft labels to adversarial examples based on their estimated proximity to the decision boundary. Instead of assigning uniform soft labels or incurring high computational overhead through explicit boundary detection, DLAT estimates soft labels through interpolation between clean and adversarial examples, substantially reducing the cost of label generation.

2. Related work

2.1. Traffic classification

Traffic classification, the task of identifying and categorizing network traffic based on application types, has evolved significantly over the years. Traditional methods such as port-based classification and payload inspection (DPI) were initially dominant but became ineffective due to dynamic port allocation, encryption, and protocol obfuscation. Statistical and machine learning-based approaches later emerged, leveraging flow-level features (e.g., packet size, inter-arrival time) to classify encrypted and unencrypted traffic. However, these methods still relied on manual feature engineering, which is time-consuming and error prone. The advent of DNNs revolutionized traffic classification by automating feature extraction and improving accuracy. Lotfollahi et al. [10] first applied deep learning to the field of traffic classification. By leveraging stacked autoencoders (SAE) and CNN architectures, it enables automatic extraction of network traffic features and achieves efficient classification of encrypted network traffic. Subsequent studies have advanced DL-based traffic classification in both accuracy and applicability. Wang et al. [19] proposed an end-to-end 1D-CNN model that processes raw packet bytes to capture spatial patterns, eliminating the need for manual feature design. Lan et al. [20] combined 1D-CNN, Bi-LSTM, and multi-head attention to classify darknet traffic, leveraging side-channel features to enhance robustness. LEXNet [21] further improved deployment efficiency by introducing a lightweight and interpretable CNN with residual connections and a prototype layer, enabling real-time inference on edge devices without sacrificing accuracy. Liu et al. [22] introduced an innovative hybrid architecture TransECA-Net, combining ECANet-enhanced CNN modules with Transformer encoders to simultaneously extract local channel-wise features and global temporal dependencies.

2.2. Adversarial example attacks and defense

While deep learning has significantly advanced traffic classification, it inherits the inherent vulnerabilities of DNNs and is susceptible to adversarial example attacks. Adversarial examples are inputs deliberately modified with subtle perturbations that cause the model to produce incorrect predictions while remaining imperceptible to human observers. This vulnerability also poses serious challenges to the security and reliability of DL-based traffic classification systems, highlighting the need for robust defense methods. Szegedy et al. [23] first revealed this weakness by formulating an optimization problem to find minimal perturbations that cause misclassification, attributing the phenomenon to local linearity in deep networks. Goodfellow et al. [15] introduced the Fast Gradient Sign Method (FGSM), which efficiently generates adversarial examples by leveraging the linear approximation of the loss function. Kurakin et al. [24] extended FGSM to an iterative version (BIM) to improve attack success. Madry et al. [17] further enhanced this with Projected Gradient Descent (PGD), adding random initialization to avoid local optima and establish a robust attack benchmark. Carlini and Wagner [25] proposed a strong optimization-based attack C&W that effectively bypasses gradient masking defenses. Sadeghzadeh [16] extends the adversarial attack to the traffic classification field and proposes adversarial pad attack and adversarial payload attack for packet and flow classification respectively, as well as adversarial burst attack for the statistical characteristics of flow time series.

Adversarial training (AT) is a widely adopted defense strategy to enhance DNNs' robustness against such adversarial attacks by incorporating adversarial examples into the training process. Proposed by Goodfellow et al. [15], AT initially used FGSM adversarial examples combined with clean examples for optimization. Madry et al. [17] showed that stronger PGD-based adversarial examples provide better robustness through a min–max optimization. However, PGD training often leads to overfitting on adversarial examples and reduced accuracy on clean data, highlighting a trade-off between robustness and generalization. To address this, Zhang et al. [26] introduced TRADES to balance this trade-off with a regularized loss. Wang et al. [27] proposed MART, which treats misclassified examples differently to enhance robustness. Dong et al. [28] developed AWP, combining input and weight perturbations to flatten the loss landscape and further reduce robust error. However, the aforementioned methods were originally proposed for image classification tasks and are not specifically designed for robust traffic classification. Directly applying these methods to traffic classification may not yield optimal results. For example, adversarial training applied to traffic data frequently induces substantial label noise, and inadequate management of such noise can considerably hinder the enhancement of model robustness.

3. Preliminaries

3.1. Pre-processing

Consider a raw network traffic flow as a discrete byte-level sequence of arbitrary length. Formally, a raw traffic flow is defined as a variable-length sequence:

$$\mathscr{F} = (b_1, b_2, \dots, b_L), \tag{1}$$

where $L \in \mathbb{N}^+$ denotes the sequence length, and each byte $b_i \in \mathbb{Z}_{256} = \{0, 1, \dots, 255\}$. The flow $\mathscr{F}$ thus resides in the input space $\mathcal{S} := \bigcup_{k=1}^{\infty} \mathbb{Z}_{256}^k$, which encompasses all finite-length byte sequences.

Following the methodology proposed by [19], each raw traffic flow $\mathscr{F}$ is standardized to a fixed length of 784 bytes to enable batch processing and compatibility with convolutional neural networks. Specifically, the transformation pipeline $\Psi : \mathcal{S} \to \mathbb{Z}_{256}^{28\times28}$ consists of:

Truncation. To standardize the size of the input dimensions of the model, we truncate the flow to the first 784 bytes:

$$\tau_k(\mathscr{F}) = (b_1, \dots, b_{\min(L,k)}), \quad k = 784. \tag{2}$$

Zero-Padding. For flows with $L < 784$, zero-padding is applied to ensure uniform dimensionality:

$$\pi_{784}(\mathscr{F}) = \begin{cases} (b_1, \dots, b_L, 0, \dots, 0) & \text{if } L < 784, \\ \tau_{784}(\mathscr{F}) & \text{otherwise.} \end{cases} \tag{3}$$

Image Mapping. The resulting 784-dimensional vector is reshaped into a 28×28 grayscale image in row-major order. We define the mapping $\Phi : \mathbb{Z}_{256}^{784} \to \mathbb{Z}_{256}^{28\times28}$ as:

$$\Phi(\mathbf{f}) = \begin{bmatrix} b_1 & b_2 & \cdots & b_{28} \\ b_{29} & b_{30} & \cdots & b_{56} \\ \vdots & \vdots & \ddots & \vdots \\ b_{745} & b_{746} & \cdots & b_{784} \end{bmatrix}, \tag{4}$$

where $\mathbf{f} = \pi_{784}(\mathscr{F})$ is the padded byte vector. This bijection arranges bytes row-by-row into a square image.

Normalization. Finally, pixel values are normalized to the range $[0, 1]$:

$$\mathcal{N}(\Phi(\mathbf{f}))_{i,j} = \frac{\Phi(\mathbf{f})_{i,j}}{255}. \tag{5}$$

The resulting tensor $x = \mathcal{N}(\Phi(\pi_{784}(\mathscr{F}))) \in [0, 1]^{28\times28}$ is used as the input to downstream neural models.

3.2. Notion

Let $\boldsymbol{x} \in [0, 1]^{28\times28}$ denote the resulting input image. The neural network takes $\boldsymbol{x}$ as input and outputs either class predictions (e.g., traffic type or application label) or binary decisions (e.g., benign vs. malicious), depending on the task. Consider a K-class classification task on the dataset $\mathcal{D} = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^N$ where $\boldsymbol{x}_i$ are preprocessed network traffic and $\boldsymbol{y}_i \in \mathcal{Y} = \{1, \dots, K\}$ are class labels. We consider a parameterized model $f_\theta : [0, 1]^{28\times28} \to \mathcal{P}$ that maps a normalized grayscale image x to a probability distribution over classes (*i.e.*, $\boldsymbol{p} = f_\theta(\boldsymbol{x})$) and the final predicted label is obtained by $\hat{\boldsymbol{y}} = \arg\max_k \boldsymbol{p}_k$. We then denote the standard loss function in the standard training process:

$$\mathcal{L}_{st}(\theta, \mathcal{D}) = \frac{1}{N} \sum_{i=1}^{N} \ell(f_\theta(\boldsymbol{x}_i), \boldsymbol{y}_i), \tag{6}$$

where N is the number of the training data, and $\ell(\cdot)$ denotes a loss function that measures the discrepancy between the model prediction and the ground-truth label (e.g., cross-entropy).

3.3. Adversarial attack

Deep learning models are known to be vulnerable to adversarial examples perturbed by imperceptible noise that induce incorrect predictions. Network traffic classifiers based on deep learning inherit this vulnerability: small, carefully designed perturbations can cause significant degradation in classification performance. Formally, given a trained model $f_\theta : [0, 1]^{28\times28} \to \mathcal{P}$ and a clean input x, an adversary aims to craft a perturbed input $x' = x + \delta$ such that:

$$\begin{aligned} &\text{Minimize} && \|\delta\|_p, \\ &\text{subject to:} && f_\theta(\boldsymbol{x} + \delta) = \boldsymbol{y}_{target}, \\ & && \boldsymbol{x} + \delta \in [0, 1]^{28\times28}, \end{aligned} \tag{7}$$

where δ denotes the adversarial perturbation and $\|\cdot\|_p$ ($p \in \{0, 1, 2, \infty\}$) quantifies perturbation magnitude. For traffic image inputs, $x' = x + \delta$ maintains the structural properties of legitimate traffic while causing misclassification. Under a white-box threat model where adversaries possess full knowledge of both the preprocessing pipeline Ψ and classifier parameters θ, attacks are executed directly in the image domain.

Crucially, the perturbation is constrained within the payload region of the traffic image, rather than the padding area.

Payload-Constrained Perturbation. To ensure semantic fidelity when mapping perturbed inputs back to the traffic domain, the adversarial perturbation δ is restricted to the non-padding (i.e., payload) region:

$$\mathcal{R} = \{(i, j) \mid 28(i-1) + j \le L\}, \tag{8}$$

where $\mathcal{R}$ denotes the set of pixels corresponding to the original L bytes of the flow $\mathscr{F}$. During attack iterations, any updates falling outside $\mathcal{R}$ are explicitly zeroed out. While this constraint does not achieve the theoretically optimal adversarial perturbation, it aligns with realistic payload limitations in network traffic and therefore produces semantically faithful perturbations that are more suitable for practical deployment. In this work, we adopt the PGD (Projected Gradient Descent) [17] as our primary adversarial method. Specifically, we perform iterative updates on the input image within the allowed perturbation budget ϵ and constrain the perturbation to the valid traffic region $\mathcal{R}$:

$$\boldsymbol{x}^{t+1} = \Pi_{\mathcal{B}_\epsilon(\boldsymbol{x}) \cap \mathcal{R}} \left(\boldsymbol{x}^t + \alpha \cdot \operatorname{sign} \left(\nabla_{\boldsymbol{x}} \mathcal{L}\left(f_\theta(\boldsymbol{x}^t), \boldsymbol{y}\right)\right)\right), \tag{9}$$

where $\mathcal{L}$ denotes the loss function, Π is the projection operator that restricts the updated input to the intersection of the valid region $\mathcal{R}$ and the ℓ_p-ball of radius ϵ centered at $\boldsymbol{x}$, and α is the step size.

3.4. Adversarial training

One of the most effective defenses against adversarial attacks is adversarial training (AT), which enhances model robustness by incorporating adversarial examples into the training process. Specifically, it formulates the training objective as a min–max optimization:

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \max_{\|\delta_i\|_p \le \epsilon} \ell\left(f_\theta(\boldsymbol{x}_i + \delta_i), \boldsymbol{y}_i\right), \tag{10}$$

For network traffic classifiers, we extend this paradigm with payload-aware constraints:

$$\min_\theta \frac{1}{N} \sum_{i=1}^{N} \max_{\delta_i \in \mathcal{C}_i} \ell(f_\theta(\boldsymbol{x}_i + \delta), \boldsymbol{y}_i) \tag{11}$$

where $\mathcal{C}_i = \left\{\delta \mid \|\delta\|_p \le \epsilon \text{ and } \delta_{(i,j)} = 0,\ \forall (i,j) \notin \mathcal{R}_i\right\}$ is the constraint set for the ith example.

4. Label noise

Label noise in adversarial training refers to the semantic mismatch between the assigned labels and the true labels of adversarial examples. As first proposed by Dong et al. [18], this phenomenon arises from the practice of assigning adversarial examples the same labels as their clean input. Given a clean input-label pair $(\boldsymbol{x}, \boldsymbol{y})$, adversarial training constructs a perturbed input $\boldsymbol{x}' = \boldsymbol{x} + \delta$ and assigns it the original label $\boldsymbol{y}$ during training. However, the true label of $\boldsymbol{x}'$ may differ due to the semantic distortion introduced by the adversarial perturbation δ. This distributional shift is especially detrimental to learning robust representations, as it misguides the optimization process.

4.1. Amplified label noise in robust traffic classification

While label noise poses a general challenge in adversarial training, it becomes even more prominent in the context of robust network traffic classification. Unlike image data, where semantic changes are often human-perceivable, traffic data is inherently opaque and lacks intuitive visual features. Consequently, different classes of traffic data are compactly distributed and highly entangled, small perturbations in the byte-level input space can lead to disproportionately large semantic changes that are not easily detectable by human inspection. In such a scenario, the probability of label mismatch between clean and adversarial examples increases. Let $\boldsymbol{x}$ be the image representation of a network flow (or packet) and $\boldsymbol{x}' = \boldsymbol{x} + \delta$ be its adversarial example. In standard adversarial training, each sample is annotated with a hard label $\boldsymbol{y}$, while the underlying ground-truth semantics are better represented by a softer distribution $\mathbb{P}(Y \mid \boldsymbol{x})$, especially for adversarial examples lying close to the decision boundary. This inherent discrepancy between the hard label and the true soft distribution can be regarded as label noise. Under adversarial perturbations $\boldsymbol{x}'$, such mismatches are further amplified, leading to a higher effective label noise rate, which we define as

$$p_e(\mathcal{D}') = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\left[\boldsymbol{y}_i \neq \arg\max \mathbb{P}(Y \mid \boldsymbol{x}_i')\right], \tag{12}$$

where $\mathcal{D}' = (\boldsymbol{x}_i', \boldsymbol{y}_i)$ denotes the adversarial training set, and $\mathbb{P}(Y \mid \boldsymbol{x}_i')$ reflects the (unknown) ground-truth label distribution of the perturbed input. Such excessive label noise disrupts the supervision learning, preventing the model from accurately learning the underlying discriminative features of the data. As a result, the classifier may overfit to incorrect labels or adversarial patterns rather than the true class semantics. This issue is particularly critical in adversarial training for traffic classification, where decision boundaries between classes are inherently subtle and highly sensitive to small perturbations.

4.2. Impact of label noise on decision boundary robustness

Adversarial training assumes that the label of an adversarial example remains unchanged from its clean example. However, when an adversarial example crosses the decision boundary into a region semantically aligned with a different class, assigning it the original label introduces semantic inconsistency. We formalize this effect in a binary classification setting. Let the input space be $\mathcal{X} \subset \mathbb{R}^d$ and the label space be $\mathcal{Y} = \{A, B\}$. Consider a classifier $f_\theta : \mathcal{X} \to [0, 1]$, where $f_\theta(\boldsymbol{x})$ denotes the predicted probability of class A, and $1 - f_\theta(\boldsymbol{x})$ is the probability of class B. The decision boundary is defined by the hypersurface $\mathcal{H}_\theta = \{\boldsymbol{x} \in \mathcal{X} \mid f_\theta(\boldsymbol{x}) = 0.5\}$. We consider an adversarial example $\boldsymbol{x}'$ generated from a clean input $\boldsymbol{x}$ of class A, such that $\boldsymbol{x}'$ lies in the classification region of class B, i.e., $f_\theta(\boldsymbol{x}') < 0.5$. During adversarial training, if $\boldsymbol{x}'$ is labeled as A (i.e., the same as $\boldsymbol{x}$), then minimizing the loss on $\boldsymbol{x}'$ pushes the decision boundary toward class B, potentially degrading the robustness of that class.

Definition 1 (*Margin Distance*)**.** Given a example $\boldsymbol{x} \in \mathcal{X}$ and a classifier $f : \mathcal{X} \to [0, 1]$, the margin distance from $\boldsymbol{x}$ to the decision boundary $\mathcal{H} = \{\boldsymbol{x} \in \mathcal{X} \mid f(\boldsymbol{x}) = 0.5\}$ is defined as:

$$dist(\boldsymbol{x}, \mathcal{H}) = \min_{\boldsymbol{x}_\mathcal{H} \in \mathcal{H}} \|\boldsymbol{x}_\mathcal{H} - \boldsymbol{x}\|_p. \tag{13}$$

Theorem 1 (*Excessive Boundary Shift Induced by Hard-Label Adversarial Training*)**.** *Consider a binary classifier $f : \mathcal{X} \to [0, 1]$, with the pre-training decision boundary defined as:*

$$\mathcal{H}_{pre} = \{\boldsymbol{x} \in \mathcal{X} \mid f_{pre}(\boldsymbol{x}) = 0.5\}. \tag{14}$$

Suppose $\boldsymbol{x}_A \in \mathcal{X}_A$ is a clean example from class A and $\boldsymbol{x}'_A = \boldsymbol{x}_A + \delta$ is an adversarial example generated to cross $\mathcal{H}_{pre}$, i.e., $f_{pre}(\boldsymbol{x}'_A) < 0.5$. Let f_{post} be the classifier obtained via hard-label adversarial training using $(\boldsymbol{x}'_A, y_A)$ as supervision, where $y_A = 1$. Then, under hard-label supervision, the training objective enforces high-confidence predictions for $\boldsymbol{x}'_A$, i.e.,

$$f_{post}(\boldsymbol{x}'_A) \gg 0.5, \tag{15}$$

which necessarily implies that the new decision boundary $\mathcal{H}_{post} = \{\boldsymbol{x} \mid f_{post}(\boldsymbol{x}) = 0.5\}$ must satisfy

$$\operatorname{dist}(\boldsymbol{x}'_A, \mathcal{H}_{post}) = \frac{f_{post}(\boldsymbol{x}'_A) - 0.5}{\|\nabla_{\boldsymbol{x}} f_{post}(\boldsymbol{x}'_A)\|_p}. \tag{16}$$

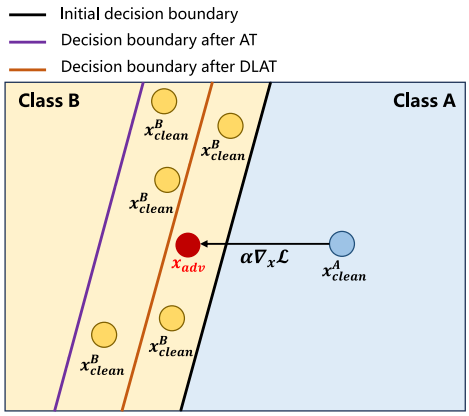


Fig. 1. Decision boundary changes: Hard-Label AT vs. Soft-Label DLAT.

In typical cases where $f_{\text{post}}(\boldsymbol{x}'_A) \to 1$, the post-training boundary moves far beyond $\boldsymbol{x}'_A$ in the direction of class B. As a result, many nearby class-B examples $\boldsymbol{x}_B \in \mathcal{X}_B$ satisfying $\boldsymbol{x}_B \approx \boldsymbol{x}'_A$ may fall into the wrong side of the decision boundary, resulting in increased misclassification. The detailed proof can be found in Appendix.

Although Theorem 1 is formulated in a binary classification setting for analytical clarity, the underlying insights naturally extend to multi-class scenarios. In the multi-class case, a classifier defines multiple decision boundaries between classes. Hard-label adversarial training on an adversarial example $\boldsymbol{x}'$ with true label y forces an increase in the logit margin:

$$z_y(\boldsymbol{x}) - z_k(\boldsymbol{x}), \quad \forall k \neq y, \tag{17}$$

which effectively pushes the decision boundaries of all other classes away from $\boldsymbol{x}'$. When $\boldsymbol{x}'$ lies near the intersection of multiple class regions, this aggressive supervision disproportionately expands the region of class y at the expense of compressing neighboring class regions, analogous to the boundary distortion shown in the binary case.

Our dynamic label assignment mitigates this issue by relaxing the overconfident supervision for adversarial examples near decision boundaries. Rather than forcing $\boldsymbol{x}'$ deep into the original decision field, the interpolated target $\boldsymbol{y}_{\text{mix}}$ the interpolated target $\boldsymbol{y}_{\text{mix}}$ guides a more appropriate adjustment of the decision boundaries. This calibrated supervision prevents the excessive boundary shift described in Theorem 1, enabling the model to maintain robustness in practical multi-class traffic classification tasks.

5. Dynamic label adversarial training

Motivated by the analysis of label noise on the robustness of adversarial training in Section 4, we propose *DLAT (Dynamic Label Adversarial Training)*, an adversarial training strategy that efficiently improves adversarial robustness utilizing dynamically mixed soft labels.

5.1. Design inspiration

In traditional adversarial training, assigning hard labels to adversarial examples introduces significant label noise, since the true label of an adversarial example may differ from its clean counterpart. This label noise forces the decision boundary to move far away from these examples, as shown in Fig. 1, ultimately leading to degraded model robustness. To address this issue, the first step is to mitigate label noise. According to Theorem 1 and Section 4.1, using soft labels can effectively reduce such label noise, thereby preventing the decision boundary from over-shifting. In binary classification, this corresponds to adjusting the boundary toward the neighborhood of the adversarial examples, which can be achieved by assigning a soft label such as $(0.5, 0.5)$ to guide adversarial training. However, in multi-classification, it is difficult to determine the soft labels of the examples near the decision boundary, and the boundary may be the intersection of decisions of multiple classes, and using soft labels such as $(\frac{1}{|\mathcal{Y}|}, \frac{1}{|\mathcal{Y}|}, \ldots, \frac{1}{|\mathcal{Y}|})$ does not fit the shape of the decision boundary well. A natural solution would be to find the examples near the current decision boundary that are within the same class as the original class of the adversarial example, and use the model's output about them as a soft label. However, explicitly detecting the decision boundary via iterative adversarial attacks is computationally expensive. Instead, DLAT capitalizes on the fact that the decision boundary must lie within the space between clean and adversarial examples, using a lightweight interpolation mechanism to approximate the soft labels of boundary-adjacent examples.

5.2. Method design

In order to accurately estimate the soft label of the examples near the decision boundary, we first need to determine the proximity of the adversarial examples to the current decision boundary, when the adversarial examples are farther away from the decision boundary, the output logits of the clean examples are given higher weight for interpolation in order to adjust the timely adjustment of the decision boundary to the vicinity of the adversarial examples, and on the contrary, the adversarial examples are given higher weight for interpolation to be able to prevent the adjusted decision boundary from crossing too much distance from the adversarial examples.

Algorithm 1: Dynamic Label Adversarial Training

Input: Network traffic dataset D; Learning rate η; Total training epochs T; Model architecture f
Initialize model f with parameters θ // Model initialization
for $i \in [T]$ **do**
  foreach *batch* $(\boldsymbol{X}, \boldsymbol{Y}) \in D$ **do**
    $\boldsymbol{X}' \leftarrow PGD(f, \boldsymbol{X}, \boldsymbol{Y})$ // Adversarial example generation
    $\boldsymbol{O} \leftarrow f(\boldsymbol{X})$
    $\boldsymbol{O}' \leftarrow f(\boldsymbol{X}')$
    $KL \leftarrow Div(\boldsymbol{O}, \boldsymbol{O}')$ // KL-based distance computation
    $\alpha \leftarrow \frac{\tanh(KL)+1}{2}$
    $\boldsymbol{Y}_{mix} \leftarrow (1-\alpha)\cdot\boldsymbol{O}' + \alpha\cdot\boldsymbol{O}$ // Mixing label construction
    $\mathcal{L}_{\text{adv}} \leftarrow Div(\boldsymbol{O}', \boldsymbol{Y}_{mix})$
    $\mathcal{L}_{\text{clean}} \leftarrow \mathcal{L}_{\text{CE}}(\boldsymbol{O}, \boldsymbol{Y})$
    $\mathcal{L}_{\text{total}} \leftarrow \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{clean}}$
    $\theta \leftarrow \theta - \eta \cdot \nabla_\theta \mathcal{L}_{\text{total}}$ // Model update
  end
end

Given a clean example $\boldsymbol{x}$ and its adversarial example $\boldsymbol{x}' = \boldsymbol{x} + \delta$, let f denote the classifier with outputs $\boldsymbol{O} = f(\boldsymbol{x})$ and $\boldsymbol{O}' = f(\boldsymbol{x}')$. Since the mapping between clean examples and hard labels can be established soon by training, we can utilize the Kullback–Leibler (KL) divergence to quantify the distance between the adversarial example and the decision boundary:

$$Div(\boldsymbol{O}, \boldsymbol{O}') = \sum_i \text{softmax}(\boldsymbol{O}_i) \log \frac{\text{softmax}(\boldsymbol{O}_i)}{\text{softmax}(\boldsymbol{O}'_i)}. \tag{18}$$

Higher Div typically indicates larger distortion and label noise. To obtain a stable and responsive mixing factor $\alpha \in [0, 1]$, we normalize $Div(\boldsymbol{O}, \boldsymbol{O}')$ using the tanh function, which provides a smooth and symmetric mapping and naturally bounds the output. Accordingly, we define:

$$\alpha = \frac{\tanh\left(Div(\boldsymbol{O}, \boldsymbol{O}')\right) + 1}{2}. \tag{19}$$

This factor interpolates between $\boldsymbol{O}'$ and $\boldsymbol{O}$ to form the mixed soft label:

$$\boldsymbol{y}_{mix} = (1-\alpha)\cdot \boldsymbol{O}' + \alpha \cdot \boldsymbol{O}. \tag{20}$$

The training objective of DLAT combines two components. The first is a KL divergence loss that aligns the model's prediction on $\boldsymbol{x}'$ with $\boldsymbol{y}_{mix}$ to improve the model robustness:

$$\mathcal{L}_{\text{adv}} = Div\left(\boldsymbol{O}', \boldsymbol{y}_{mix}\right), \tag{21}$$

where the second is a cross-entropy loss that is used to allow the model to learn generalization knowledge and improve clean example classification accuracy:

$$\mathcal{L}_{\text{clean}} = -\sum_i \boldsymbol{y}_i \log \text{softmax}(\boldsymbol{O}_i). \tag{22}$$

The overall loss is formulated as:

$$\min_{\theta} \max_{\delta_i \in C_i} \left[\mathcal{L}_{\text{adv}}(f_\theta(\boldsymbol{x}+\delta), \boldsymbol{y}_{mix}) + \mathcal{L}_{\text{clean}}(f_\theta(\boldsymbol{x}), \boldsymbol{y})\right]. \tag{23}$$

By dynamically adapting label softness based on Eq. (18)–(20) and balancing loss components Eq. (21)–(23), DLAT mitigates excessive boundary shift caused by label noise, enabling models to learn robust decision boundaries for tasks like traffic classification. The pseudo-code for DLAT is presented on Algorithm 1.

6. Experiments

In this section, we perform a wide variety of comprehensive experiments to evaluate the performance of DLAT on both clean and adversarial traffic. These evaluations are carried out on two datasets and compared against four state-of-the-art adversarial training methods in the computer vision field.

6.1. Experiment setup

Datasets. Experiments are performed using the ISCX VPN-nonVPN dataset [29] and the CICIoT2022 dataset [30]. The former includes encrypted and unencrypted traffic, while the latter focuses on IoT-related scenarios with both benign and malicious behaviors. We construct three experimental settings from those datasets. The first, referred to as ISCX-VPN, includes six categories of encrypted VPN traffic: VPN_Chat, VPN_Email, VPN_File Transfer, VPN_P2P, VPN_Streaming, and VPN_VoIP. The second setting, named ISCX-ALL, expands the classification scope to twelve categories by incorporating six VPN and six non-VPN traffic types. The third setting, derived from the CICIoT2022 dataset, defines a six-class classification task encompassing typical IoT device states and activities. The categories include: Power, Idle, Interactions, Scenarios, Active, and Attacks. Since the original datasets exhibit significant class imbalance, we first split the data into training and testing sets with a 9:1 ratio, and then apply class-wise balancing separately within each subset to ensure a relatively balanced class distribution. The statistics of the balanced datasets are summarized in Table 1, 2 and 3.

Training. We adopt two representative neural network architectures as backbone models: PreActResNet [31], DenseNet [32], MobileNet [33], WideResNet [34], and FFNN (Feed-Forward Neural Network) [35]. Both models are trained for 80 epochs using the momentum-based stochastic gradient descent (MSGD) [36], with a momentum coefficient of 0.9 and a weight decay of 5×10^{-4}. The initial learning rate is set to 0.1, and a multi-stage learning rate decay strategy is applied: the learning rate is reduced by a factor of 10 at the 40th epoch.

Attack and defense settings. For adversarial evaluation, we adopt the widely used PGD-20 under the ℓ_∞ norm constraint. The perturbation radius ϵ is set to 24/255, and the step size α is 4/255. For generating adversarial examples used in adversarial training, we employ PGD-10 under the same ℓ_∞-bounded perturbation settings.

Table 1
The balanced ISCX-VPN dataset.

Type	Imbalanced dataset	Imbalanced dataset	
	Total number	Training set number	Test set number
VPN_Chat	7946	1500	200
VPN_Email	596	1500	59
VPN_File Transfer	1898	1500	189
VPN_P2P	912	1500	91
VPN_Streaming	1199	1500	119
VPN_VoIP	20 581	1500	200

Table 2
The balanced CICIoT2022 dataset.

Type	Imbalanced dataset	Imbalanced dataset	
	Total number	Training set number	Test set number
VPN_Chat	7946	1500	200
VPN_Email	596	1500	59
VPN_File Transfer	1898	1500	189
VPN_P2P	912	1500	91
VPN_Streaming	1199	1500	119
VPN_VoIP	20 581	1500	200

Table 3
The balanced ISCX-ALL dataset.

Type	Imbalanced dataset	Imbalanced dataset	
	Total number	Training set number	Test set number
Chat	7681	5400	600
Email	6459	5400	600
File Transfer	7405	5400	600
P2P	1849	1652	184
Streaming	3936	3540	393
VoIP	19 597	5400	600
VPN_Chat	7946	5400	600
VPN_Email	596	538	59
VPN_File Transfer	1898	1754	189
VPN_P2P	912	830	91
VPN_Streaming	1199	1108	119
VPN_VoIP	20 581	5400	600

Evaluation Metrics. In our experiments, we adopt two primary evaluation metrics to assess the effectiveness of DLAT: the *Robust Classification Accuracy* (RCC) and the *Clean Sample Accuracy* (ACC). ASR measures the proportion of adversarial traffic that successfully fools the model, indicating the robustness of the defense mechanism under adversarial attacks. A lower RCC implies stronger robustness. In contrast, ACC evaluates the classification accuracy on clean, unperturbed traffic, reflecting the model's predictive performance under normal conditions. A higher ACC indicates better generalization and utility in benign settings. We report both metrics to provide a comprehensive assessment of the trade-off between robustness and standard accuracy.

Baselines. We compare DLAT to the following representative adversarial training baselines, including PGD-AT [17], TRADES [26], MART [27], and AWP [28]. All baseline methods are implemented following their original settings. For TRADES, the trade-off parameter λ is set to 1/6, as suggested in the original paper. For AWP, the weight perturbation step size γ is set to 0.01. Unlike those training methods, which still rely on hard labels and thus remain sensitive to mislabeled data, DLAT explicitly incorporates soft-label supervision, making it more robust under label noise.

6.2. The effectiveness of DLAT

Clean accuracy assessment. As shown in Table 4, the normal model trained without adversarial defenses achieves the highest ACC across

Table 4
The clean sample accuracy (ACC) and robust classification accuracy (RCC) of different adversarial training methods across four network architectures: ResNet, DenseNet, MobileNet, WideResNet, and FFNN on the ISCX-VPN, ISCX-ALL and CICIoT2022 datasets (%).

Dataset	Method	Model									
		ResNet		DenseNet		MobileNet		WideResNet		FFNN	
		ACC	RCC	ACC	RCC	ACC	RCC	ACC	RCC	ACC	RCC
ISCX-VPN	Normal	99.02 ± 0.30	0.00 ± 0.00	99.92 ± 0.08	0.67 ± 0.09	99.17 ± 0.00	3.58 ± 0.14	99.75 ± 0.00	0.83 ± 0.07	98.25 ± 0.00	7.67 ± 0.58
	PGD-AT	98.72 ± 0.18	96.32 ± 0.29	96.02 ± 0.23	91.00 ± 0.72	97.87 ± 0.25	90.00 ± 2.69	**99.35 ± 0.08**	96.01 ± 0.11	**97.25 ± 0.24**	87.00 ± 0.81
	TRADES	96.75 ± 0.37	94.62 ± 0.30	92.98 ± 0.29	89.92 ± 0.15	93.18 ± 0.44	85.35 ± 3.38	97.92 ± 0.24	96.03 ± 0.18	92.02 ± 0.41	83.68 ± 0.87
	MART	98.08 ± 0.43	94.20 ± 0.59	82.65 ± 0.72	78.90 ± 0.53	80.83 ± 1.76	70.85 ± 1.74	98.51 ± 0.19	92.72 ± 0.17	93.28 ± 0.20	84.58 ± 0.60
	AWP	98.18 ± 0.17	96.22 ± 0.17	95.40 ± 0.33	92.92 ± 0.09	93.40 ± 0.42	90.10 ± 0.49	73.82 ± 0.46	72.18 ± 0.54	95.63 ± 0.24	**88.32 ± 0.29**
	DLAT	**98.83 ± 0.09**	**96.53 ± 0.08**	**98.77 ± 0.26**	**93.93 ± 0.42**	**98.20 ± 0.10**	**93.07 ± 0.47**	99.08 ± 0.05	**96.38 ± 0.36**	96.88 ± 0.17	86.37 ± 0.30
ISCX-ALL	Normal	93.95 ± 4.36	2.04 ± 1.06	96.70 ± 2.11	0.23 ± 0.07	91.52 ± 4.99	3.74 ± 0.12	96.22 ± 1.48	7.23 ± 0.48	88.48 ± 0.27	1.61 ± 0.21
	PGD-AT	88.56 ± 0.10	**87.34 ± 0.20**	82.96 ± 0.26	80.61 ± 0.30	82.19 ± 0.24	**78.87 ± 0.73**	88.63 ± 0.03	86.12 ± 2.89	83.00 ± 0.34	**77.23 ± 0.29**
	TRADES	88.31 ± 0.13	86.19 ± 0.45	79.19 ± 1.12	73.98 ± 3.39	80.39 ± 0.80	75.26 ± 2.93	87.32 ± 1.41	84.90 ± 2.54	76.47 ± 1.90	71.01 ± 0.75
	MART	88.19 ± 0.18	86.33 ± 0.51	77.22 ± 0.19	76.08 ± 0.22	80.78 ± 0.33	77.79 ± 0.31	87.67 ± 0.12	86.10 ± 0.45	75.99 ± 0.64	69.95 ± 1.79
	AWP	86.31 ± 0.11	85.44 ± 0.10	78.00 ± 0.19	76.43 ± 0.48	78.83 ± 0.07	77.58 ± 0.16	85.85 ± 0.12	84.71 ± 0.05	81.30 ± 0.21	76.91 ± 0.21
	DLAT	**89.44 ± 0.32**	86.68 ± 0.40	**88.83 ± 0.80**	**82.18 ± 0.43**	**84.35 ± 0.36**	75.84 ± 1.27	**88.71 ± 0.02**	**87.14 ± 0.41**	**86.79 ± 0.26**	74.32 ± 0.81
CICIoT2022	Normal	99.82 ± 0.32	0.04 ± 0.01	99.73 ± 0.01	0.63 ± 0.02	98.50 ± 2.59	0.00 ± 0.00	99.99 ± 0.00	0.56 ± 0.01	99.67 ± 0.06	0.12 ± 0.06
	PGD-AT	99.27 ± 0.08	96.26 ± 3.18	98.20 ± 0.02	96.86 ± 0.44	**98.20 ± 0.79**	97.65 ± 0.47	99.46 ± 0.21	93.73 ± 0.46	83.32 ± 2.40	81.36 ± 2.58
	TRADES	98.35 ± 0.82	98.90 ± 0.57	98.04 ± 0.00	**97.81 ± 1.36**	98.05 ± 0.31	91.38 ± 0.74	98.06 ± 0.02	97.62 ± 0.19	96.84 ± 0.11	89.20 ± 0.27
	MART	98.19 ± 0.02	96.37 ± 2.27	98.05 ± 0.31	95.50 ± 0.50	98.06 ± 0.28	95.20 ± 0.40	99.00 ± 0.05	97.00 ± 0.10	98.20 ± 0.20	91.28 ± 1.50
	AWP	98.25 ± 0.10	96.50 ± 0.20	98.10 ± 0.15	96.00 ± 0.25	98.15 ± 0.12	95.50 ± 0.30	99.10 ± 0.05	98.00 ± 0.10	98.00 ± 0.15	90.10 ± 0.50
	DLAT	**99.70 ± 0.02**	**99.20 ± 0.12**	**98.89 ± 0.17**	97.12 ± 0.24	98.06 ± 0.28	**97.88 ± 0.14**	**99.66 ± 0.02**	**98.99 ± 0.11**	**98.87 ± 0.09**	**91.93 ± 0.86**

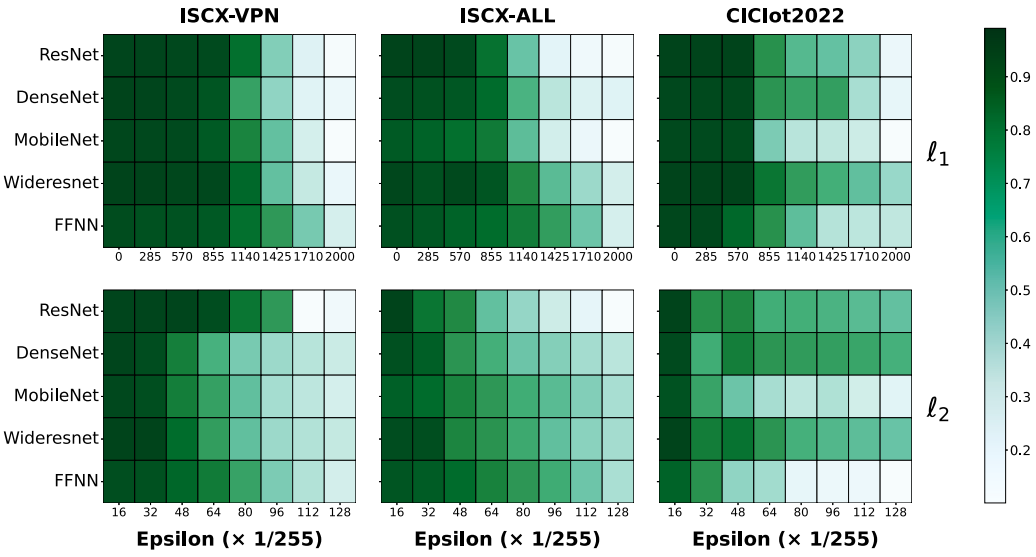


Fig. 2. The robust classification accuracy (RCC) of DLAT under ℓ_1 and ℓ_2 norm-bounded PGD-20 attacks on datasets ISCX-VPN, ISCX-ALL and CICIot2022.

all architectures, ranging from 98.25% to 99.92% on ISCX-VPN, from 88.48% to 96.70% on ISCX-ALL, and from 98.50% to 99.99% on CICIot2022. However, it fails completely under adversarial attacks, with robustness classification accuracy (RCC) close to zero. In the table, boldface highlights the best performance for each metric, while underlining indicates the second-best. Compared to the normal model, adversarial training methods such as PGD-AT, TRADES, and MART significantly improve robustness, albeit at the cost of decreased clean accuracy. Specifically, PGD-AT maintains relatively higher ACC (*e.g.*, 98.72% on ResNet and 88.56% on ISCX-ALL, while TRADES and MART show larger reductions in ACC on clean examples). Our method, DLAT, consistently achieves competitive ACC, reaching up to 98.83% on ResNet and 89.44% on ISCX-ALL, surpassing all baselines on ISCX-ALL and maintaining top-tier accuracy on ISCX-VPN and CICIot2022. These results demonstrate that DLAT effectively enhances robustness with minimal compromise to clean performance.

Robust accuracy assessment. We first evaluate the RCC of various adversarial training methods under adversarial attacks. As shown in Table 4, adversarial training markedly improves RCC compared with the normal model, which exhibits near-zero robustness. Among the compared methods, DLAT consistently surpasses most baselines in the majority of cases across both datasets and network architectures. Specifically, on ISCX-VPN, DLAT attains RCC scores above 86% across all architectures, notably outperforming PGD-AT, TRADES, MART, and AWP, with top results exceeding 96% on ResNet and WideResNet. Similarly, on ISCX-ALL and CICIot2022, it maintains leading robustness, achieving up to 87.14% and 98.99% RCC on WideResNet and surpassing competing methods by a clear margin. These findings underscore the superior robustness of DLAT while retaining competitive clean accuracy.

Secondly, to further assess the robustness of DLAT against unseen adversarial threats, we evaluate its robustness under a diverse set of attack methods, including adversarial perturbations constrained by different norm bounds (*i.e.*, ℓ_1 and ℓ_2 norms) as well as FGSM [15], PGD-100 [17], and AutoAttack [37]. We first report the performance of DLAT under ℓ_1- and ℓ_2-bounded PGD-20 attacks on the ISCX-VPN, ISCX-ALL, and CICIot2022 datasets, as illustrated in Fig. 2. Each heatmap visualizes the RCC achieved by five different models under increasing perturbation radii. It can be observed that DLAT exhibits strong robustness under both ℓ_1- and ℓ_2-bounded PGD-20 attacks. Notably, the defense is more effective against ℓ_1-norm perturbations, as indicated by the overall darker color tones in the corresponding heatmaps. This suggests that DLAT better preserves classification performance when facing sparse but high-magnitude perturbations. Among the evaluated models, ResNet and DenseNet generally exhibit higher RCC scores across both norm types and datasets, with RCC remaining above 0.8 under moderate ℓ_1 perturbations (e.g., ϵ =

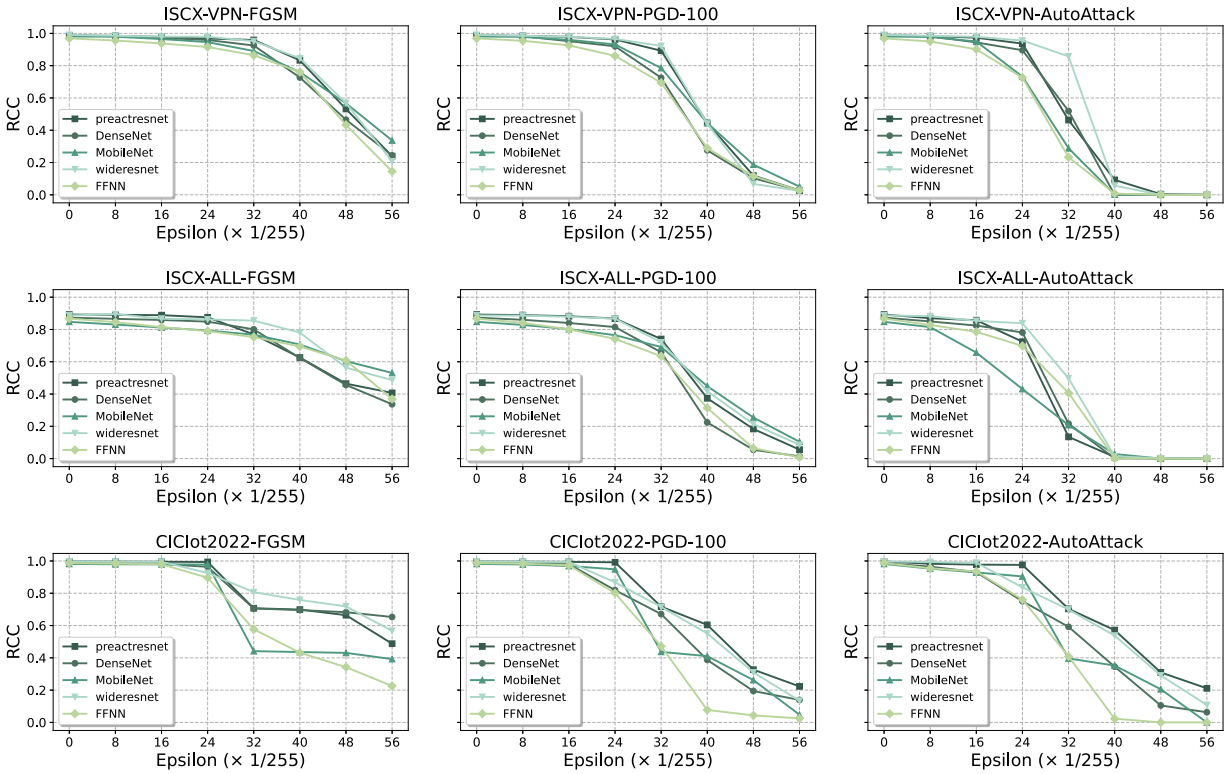


Fig. 3. The RCC of DLAT under FGSM, PGD-100, AutoAttack on ISCX-VPN, ISCX-ALL, and CICIot2022 datasets.

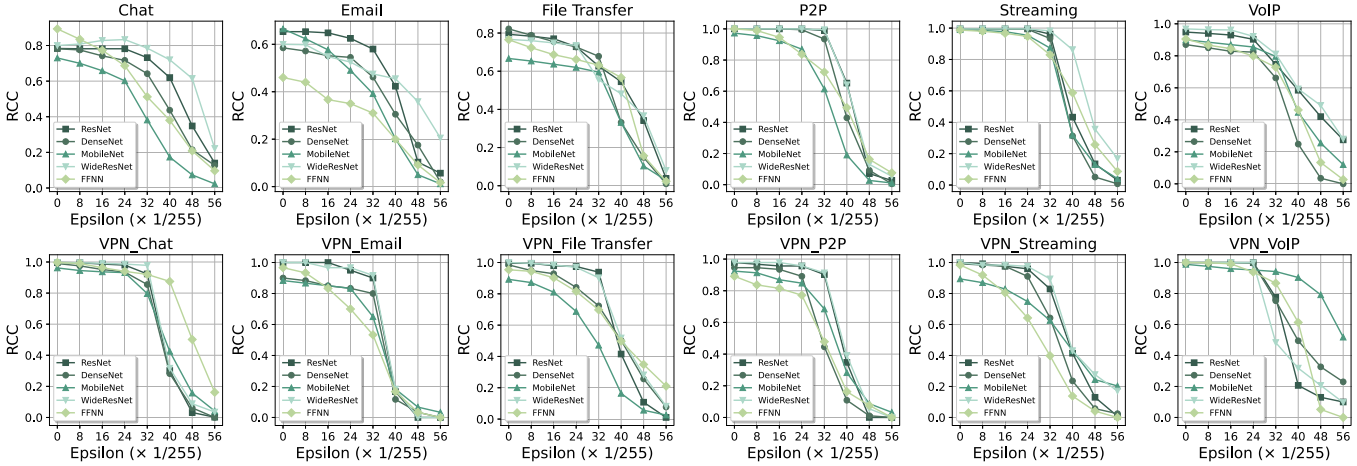
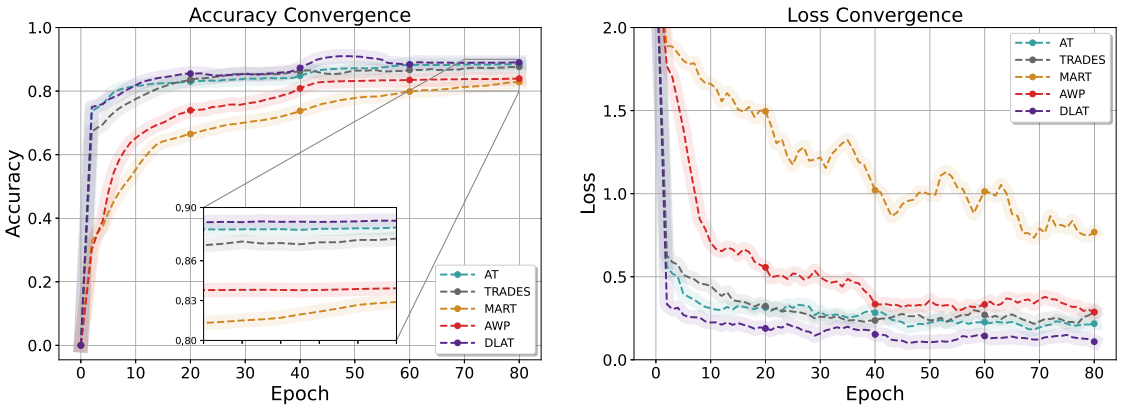


Fig. 4. The robust classification accuracy (RCC) of various models across classes on ISCX-ALL under increasing adversarial perturbation radii.

1140/255). In contrast, MobileNet and DenseNet show relatively lower robustness, particularly under ℓ_2-bounded attacks, where RCC values gradually decrease below 0.6 as the perturbation radius increases. Nonetheless, the performance degradation across all models is smooth rather than abrupt, suggesting that DLAT retains a degree of robustness and stability.

As shown in Fig. 3, we further assess the performance of DLAT under three previously unseen adversarial attacks: FGSM, PGD-100, and AutoAttack. Under FGSM, all evaluated models exhibit strong robustness, with RCC values typically exceeding 0.85 below $\epsilon = 24/255$, and models such as ResNet and WideResNet experiencing only marginal performance degradation. As the perturbation strength increases under PGD-100, the RCC gradually decreases across all models. Nonetheless, most models achieve RCCs above 0.5 at $\epsilon = 32/255$ on the ISCX-VPN dataset, indicating a moderate level of robustness. AutoAttack presents the most challenging scenario, leading to a more pronounced decline in performance, particularly when ϵ exceeds 24/255. Despite this, architectures such as ResNet and wideresnet continue to maintain RCC above 0.5 at $\epsilon = 32/255$, suggesting that DLAT remains effective even under adaptive and high-strength adversarial attacks. These results collectively demonstrate the generalization capability of the framework across a broad range of attacks and perturbation intensities.

We thirdly evaluate the robustness of DLAT under varying attack intensities, where the attack intensity corresponds to the radii of adversarial perturbations (denoted by Epsilon ϵ). As comprehensively illustrated in Fig. 4, we present the RCC performance for each individual class within the ISCX-ALL dataset (including Chat, Email, File Transfer, P2P, Streaming, VoIP, VPN_Chat, VPN_Email, VPN_File Transfer, VPN_P2P, VPN_Streaming, and VPN_VoIP) across multiple network architectures (ResNet, DenseNet, MobileNet, WideResNet, FFNN) under increasing perturbation radii (ϵ ranging from 0 to 56/255). The adversarial training of DLAT is performed using adversarial examples



(a) Accuracy curve (b) Loss curve

Fig. 5. Comparison of accuracy and loss convergence results for DenseNet on the ISCX-ALL Dataset.

generated with a perturbation radius of $\epsilon = 24/255$. As shown in Fig. 4, across most classes and architectures, the trained models demonstrate strong robustness when the attack intensity remains within or below this radius ($\epsilon \leq 24/255$), and the models still maintain relatively strong resilience to perturbations (i.e., $24/255 < \epsilon < 32/255$). However, once ϵ exceeds $32/255$, the attack becomes significantly stronger, leading to a noticeable drop in RCC, especially for non-VPN classes.

6.3. The efficiency of DLAT

To evaluate the training efficiency of DLAT, we compare its convergence with that of representative adversarial training baselines, including AT, TRADES, MART, and AWP. As illustrated in Fig. 5, DLAT demonstrates significantly faster convergence in both accuracy and loss. Specifically, in the accuracy curve (Fig. 5(a), DLAT rapidly improves during the initial training epochs, reaching a stable accuracy above 0.85 within 30 epochs. In contrast, competing methods exhibit slower convergence and lower final performance, with TRADES and MART stabilizing below 0.80. Similarly, the loss curve (Fig. 5(b) further highlights the advantage of DLAT in optimization stability. It consistently maintains a lower loss value throughout training and converges to a final loss below 0.3, which is noticeably lower than those of other methods. These results collectively demonstrate that DLAT not only accelerates the convergence process but also facilitates optimization toward better minima, indicating its efficiency and practicality for robust model training.

In addition to its fast convergence, DLAT maintains comparable training time per epoch to other adversarial training methods, as reported in Table 5. Across different model architectures and datasets, the time cost of DLAT remains close to that of AT, TRADES, MART, and AWP. By achieving improved robustness and faster convergence without sacrificing efficiency, DLAT offers a practical solution for robust network traffic classification.

7. Conclusion

In this paper, we investigated the vulnerability of deep traffic classifiers to adversarial examples and the label noise introduced by hard-label supervision in adversarial training. To address this issue, we proposed DLAT, a dynamic adversarial training framework that assigns soft labels to adversarial examples based on the similarity between clean and perturbed outputs. This similarity-guided interpolation helps mitigate label noise and align the decision boundary more effectively. Experimental results on traffic classification benchmarks demonstrate that DLAT consistently improves robustness and generalization over standard adversarial training.

Table 5
Comparison of the time consumption for each epoch of the adversarial training methods (s).

Dataset	Model	AT	TRADES	MART	AWP	DLAT
ISCX-VPN	ResNet	16.99	17.98	19.38	19.19	19.07
	DenseNet	12.59	14.02	14.52	15.84	14.28
	MobileNet	26.14	28.55	28.14	30.83	27.98
	WideResNet	139.62	136.84	147.27	140.37	152.07
	FFNN	4.02	3.85	3.94	4.36	4.41
ISCX-ALL	ResNet	74.32	80.69	84.49	89.11	81.57
	DenseNet	57.64	60.83	63.62	66.78	62.95
	MobileNet	113.71	114.23	130.42	129.99	117.19
	WideResNet	673.35	621.27	688.85	688.37	762.18
	FFNN	16.43	15.03	17.86	17.62	16.31
CICIoT2022	ResNet	47.35	48.92	51.19	51.32	49.63
	DenseNet	61.02	63.11	66.68	68.92	64.90
	MobileNet	121.56	122.91	132.23	135.13	124.87
	WideResNet	680.37	690.82	703.16	710.55	695.09
	FFNN	18.06	19.42	18.98	19.56	20.43

CRediT authorship contribution statement

Haoyu Tong: Writing – original draft. **Meixia Miao:** Methodology, Formal analysis, Project administration. **Yundong Liu:** Data curation. **Xiaoyu Zhang:** Writing – original draft, Supervision. **Xiangyang Luo:** Resources, Funding acquisition. **Willy Susilo:** Visualization, Validation, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is funded by the Open Foundation of Key Laboratory of Cyberspace Security, Ministry of Education of China and Henan Key Laboratory of Cyberspace Situation Awareness (No. KLCS20240103), National Natural Science Foundation of China (No. 62472345), and Fundamental Research Funds for the Central Universities, China (No. QTZX25088).

Appendix. The proof Theorem 1

Theorem 1 (*Excessive Boundary Shift Induced by Hard-Label Adversarial Training*). *Consider a binary classifier* $f : \mathcal{X} \to [0,1]$, *with the pre-training decision boundary defined as:*

$$\mathcal{H}_{pre} = \{\boldsymbol{x} \in \mathcal{X} \mid f_{pre}(\boldsymbol{x}) = 0.5\}.$$

Suppose $\boldsymbol{x}_A \in \mathcal{X}_A$ *is a clean example from class A and* $\boldsymbol{x}'_A = \boldsymbol{x}_A + \delta$ *is an adversarial example generated to cross* $\mathcal{H}_{pre}$, *i.e.,* $f_{pre}(\boldsymbol{x}'_A) < 0.5$. *Let* f_{post} *be the classifier obtained via hard-label adversarial training using* $(\boldsymbol{x}'_A, y_A)$ *as supervision, where* $y_A = 1$. *Then, under hard-label supervision, the training objective enforces high-confidence predictions for* $\boldsymbol{x}'_A$, *i.e.,*

$$f_{post}(\boldsymbol{x}'_A) \gg 0.5,$$

which necessarily implies that the new decision boundary $\mathcal{H}_{post} = \{\boldsymbol{x} \mid f_{post}(\boldsymbol{x}) = 0.5\}$ *must satisfy*

$$\mathrm{dist}(\boldsymbol{x}'_A, \mathcal{H}_{post}) = \frac{f_{post}(\boldsymbol{x}'_A) - 0.5}{\|\nabla_{\boldsymbol{x}} f_{post}(\boldsymbol{x}'_A)\|_p}.$$

Proof. Let $\boldsymbol{x}_A \in \mathcal{X}_A$ be a clean example correctly classified as class A, and let $\boldsymbol{x}'_A = \boldsymbol{x}_A + \delta$ be its adversarial variant generated to cross the original decision boundary $\mathcal{H}_{\mathrm{pre}}$, i.e.,

$$f_{\mathrm{pre}}(\boldsymbol{x}'_A) < 0.5.$$

Hard-label adversarial training uses the tuple $(\boldsymbol{x}'_A, y_A = 1)$ as supervised data, forcing the model f_{post} to assign high confidence to $\boldsymbol{x}'_A$:

$$f_{\mathrm{post}}(\boldsymbol{x}'_A) \to 1.$$

Now, consider the new decision boundary:

$$\mathcal{H}_{\mathrm{post}} = \{\boldsymbol{x} \mid f_{\mathrm{post}}(\boldsymbol{x}) = 0.5\}.$$

We approximate f_{post} in a neighborhood of $\boldsymbol{x}'_A$ using a first-order Taylor expansion:

$$f_{\mathrm{post}}(\boldsymbol{x}) \approx f_{\mathrm{post}}(\boldsymbol{x}'_A) + \nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)^\top (\boldsymbol{x} - \boldsymbol{x}'_A).$$

Let $\boldsymbol{x}_{\mathcal{H}} \in \mathcal{H}_{\mathrm{post}}$ denote the closest point on the new boundary to $\boldsymbol{x}'_A$. By definition,

$$f_{\mathrm{post}}(\boldsymbol{x}_{\mathcal{H}}) = 0.5.$$

Using the linear approximation, we have:

$$0.5 \approx f_{\mathrm{post}}(\boldsymbol{x}'_A) + \nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)^\top (\boldsymbol{x}_{\mathcal{H}} - \boldsymbol{x}'_A).$$

Solving for the shift vector:

$$\nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)^\top (\boldsymbol{x}_{\mathcal{H}} - \boldsymbol{x}'_A) \approx 0.5 - f_{\mathrm{post}}(\boldsymbol{x}'_A).$$

Let $\boldsymbol{v} = \nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)/\|\nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)\|_p$ be the normalized gradient (i.e., the local normal direction to the decision boundary). Then the minimal distance from $\boldsymbol{x}'_A$ to the boundary is:

$$\|\boldsymbol{x}_{\mathcal{H}} - \boldsymbol{x}'_A\|_p = \frac{|f_{\mathrm{post}}(\boldsymbol{x}'_A) - 0.5|}{\|\nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)\|_p}.$$

As $f_{\mathrm{post}}(\boldsymbol{x}'_A) \to 1$, this implies:

$$\mathrm{dist}(\boldsymbol{x}'_A, \mathcal{H}_{\mathrm{post}}) \to \frac{0.5}{\|\nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)\|_p}.$$

This lower bound quantifies how far the decision boundary must move beyond $\boldsymbol{x}'_A$ to satisfy $f_{\mathrm{post}}(\boldsymbol{x}'_A) = 1$. If $\nabla_{\boldsymbol{x}} f_{\mathrm{post}}(\boldsymbol{x}'_A)$ is not vanishingly large, this distance is significant. Finally, since $\boldsymbol{x}'_A$ was crafted to lie just beyond $\mathcal{H}_{\mathrm{pre}}$, i.e., in close proximity to the original boundary, the boundary movement beyond $\boldsymbol{x}'_A$ implies that the new decision boundary has crossed deep into the region previously occupied by class B. Therefore, class-B examples in the vicinity of $\boldsymbol{x}'_A$ are likely to be misclassified as class A under f_{post}. □

Data availability

Data will be made available on request.

References

[1] A. Azab, M. Khasawneh, S. Alrabaee, K.-K.R. Choo, M. Sarsour, Network traffic classification: Techniques, datasets, and challenges, Digit. Commun. Netw. 10 (3) (2024) 676–692.

[2] H. Yuan, G. Li, A survey of traffic prediction: from spatio-temporal data to intelligent transportation, Data Sci. Eng. 6 (1) (2021) 63–85.

[3] A.W. Moore, K. Papagiannaki, Toward the accurate identification of network applications, in: International Workshop on Passive and Active Network Measurement, Springer, 2005, pp. 41–54.

[4] A. Madhukar, C. Williamson, A longitudinal study of P2P traffic classification, in: 14th IEEE International Symposium on Modeling, Analysis, and Simulation, IEEE, 2006, pp. 179–188.

[5] S. Fernandes, R. Antonello, T. Lacerda, A. Santos, D. Sadok, T. Westholm, Slimming down deep packet inspection systems, in: IEEE INFOCOM Workshops 2009, IEEE, 2009, pp. 1–6.

[6] N. Hubballi, M. Swarnkar, M. Conti, BitProb: Probabilistic bit signatures for accurate application identification, IEEE Trans. Netw. Serv. Manag. 17 (3) (2020) 1730–1741, http://dx.doi.org/10.1109/TNSM.2020.2999856.

[7] A. Azab, P. Watters, R. Layton, Characterising network traffic for skype forensics, in: 2012 Third Cybercrime and Trustworthy Computing Workshop, 2012, pp. 19–27, http://dx.doi.org/10.1109/CTC.2012.14.

[8] H. Mohajeri Moghaddam, Skypemorph: Protocol Obfuscation for Censorship Resistance, University of Waterloo, 2013.

[9] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[10] M. Lotfollahi, M.J. Siavoshani, R.S.H. Zade, M. Saberian, Deep packet: a novel approach for encrypted traffic classification using deep learning, Soft Comput. 24 (2017) 1999–2012, URL https://api.semanticscholar.org/CorpusID:35187639.

[11] L. Yang, A. Finamore, F. Jun, D. Rossi, Deep learning and traffic classification: Lessons learned from a commercial-grade dataset with hundreds of encrypted and zero-day applications, 2021, arXiv preprint arXiv:2104.03182.

[12] M.H. Pathmaperuma, Y. Rahulamathavan, S. Dogan, A.M. Kondoz, Deep learning for encrypted traffic classification and unknown data detection, Sensors 22 (19) (2022) 7643.

[13] X. Lin, G. Xiong, G. Gou, Z. Li, J. Shi, J. Yu, Et-bert: A contextualized datagram representation with pre-training transformers for encrypted traffic classification, in: Proceedings of the ACM Web Conference 2022, 2022, pp. 633–642.

[14] X. Ma, W. Zhu, J. Wei, Y. Jin, D. Gu, R. Wang, EETC: An extended encrypted traffic classification algorithm based on variant resnet network, Comput. Secur. 128 (2023) 103175.

[15] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, ICLR, 2014.

[16] A.M. Sadeghzadeh, S. Shiravi, R. Jalili, Adversarial network traffic: Towards evaluating the robustness of deep-learning-based network traffic classification, IEEE Trans. Netw. Serv. Manag. 18 (2) (2021) 1962–1976.

[17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, ICLR, 2018.

[18] C. Dong, L. Liu, J. Shang, Label noise in adversarial training: A novel perspective to study robust overfitting, Adv. Neural Inf. Process. Syst. 35 (2022) 17556–17567.

[19] W. Wang, M. Zhu, J. Wang, X. Zeng, Z. Yang, End-to-end encrypted traffic classification with one-dimensional convolution neural networks, in: 2017 IEEE International Conference on Intelligence and Security Informatics, ISI, IEEE, 2017, pp. 43–48.

[20] J. Lan, X. Liu, B. Li, Y. Li, T. Geng, DarknetSec: A novel self-attentive deep learning method for darknet traffic classification and application identification, Comput. Secur. 116 (2022) 102663.

[21] K. Fauvel, F. Chen, D. Rossi, A lightweight, efficient and explainable-by-design convolutional neural network for internet traffic classification, in: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2023, pp. 4013–4023.

[22] Z. Liu, Y. Xie, Y. Luo, Y. Wang, X. Ji, TransECA-net: A transformer-based model for encrypted traffic classification, Appl. Sci. 15 (6) (2025) 2977.

[23] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, 2013, arXiv:1312.6199.

[24] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: Artificial Intelligence Safety and Security, Chapman and Hall/CRC, 2018, pp. 99–112.

[25] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy, S&P, IEEE, 2017, pp. 39–57.

[26] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, M. Jordan, Theoretically principled trade-off between robustness and accuracy, in: International Conference on Machine Learning, PMLR, 2019, pp. 7472–7482.

[27] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, Q. Gu, Improving adversarial robustness requires revisiting misclassified examples, in: International Conference on Learning Representations, ICLR, 2019.
[28] D. Wu, S.-T. Xia, Y. Wang, Adversarial weight perturbation helps robust generalization, Adv. Neural Inf. Process. Syst. 33 (2020) 2958–2969.
[29] G.D. Gil, A.H. Lashkari, M. Mamun, A.A. Ghorbani, Characterization of encrypted and VPN traffic using time-related features, in: Proceedings of the 2nd International Conference on Information Systems Security and Privacy, ICISSP 2016, SciTePress Setúbal, Portugal, 2016, pp. 407–414.
[30] S. Dadkhah, H. Mahdikhani, P.K. Danso, A. Zohourian, K.A. Truong, A.A. Ghorbani, Towards the development of a realistic multidimensional IoT profiling dataset, in: 2022 19th Annual International Conference on Privacy, Security & Trust, PST, IEEE, 2022, pp. 1–11.
[31] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, in: Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, the Netherlands, October 11–14, 2016, Proceedings, Part IV 14, Springer, 2016, pp. 630–645.
[32] G. Huang, Z. Liu, L. Van Der Maaten, K.Q. Weinberger, Densely connected convolutional networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4700–4708.
[33] A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, H. Adam, Mobilenets: Efficient convolutional neural networks for mobile vision applications, 2017, arXiv preprint arXiv:1704.04861.
[34] S. Zagoruyko, N. Komodakis, Wide residual networks, 2016, arXiv preprint arXiv:1605.07146.
[35] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning representations by back-propagating errors, Nature 323 (6088) (1986) 533–536.
[36] N. Qian, On the momentum term in gradient descent learning algorithms, Neural Netw. 12 (1) (1999) 145–151.
[37] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: ICML, 2020.