Robust And Explainable Deep Learning Framework Against Adversarial Attacks For Network Threat Classification

Ruili Wang*

Puyang Petrochemical Vocational and Technical College, 457000, Puyang China

* *Corresponding author. E-mail: rayliwong@126.com*

Received: Apr. 19, 2026; Accepted: May. 18, 2026

With the rapid digitization of critical infrastructure and the proliferation of complex cyber threats, deep learning (DL) has become a core technology for network threat classification, enabling accurate identification of malicious activities such as DDoS attacks, malware infections, and phishing attempts. However, DL models are inherently vulnerable to adversarial attacks, subtle, human-imperceptible perturbations added to input network traffic features that can misleadingly alter model predictions, posing severe risks to network security. Additionally, the black-box nature of most advanced DL models (e.g., Transformers, deep neural networks) hinders their practical deployment in security-critical scenarios, as security analysts cannot interpret the reasoning behind classification decisions. To address these two critical challenges with robustness against adversarial attacks and model explainability, this paper proposes a novel Robust and Explainable Deep Learning Framework (REDLF) for network threat classification. The framework integrates three core components: (1) an Adversarial Training Module (ATM) based on projected gradient descent (PGD) and physical environment modeling to enhance model robustness; (2) a Feature Enhancement Module (FEM) that combines attention mechanisms and variational autoencoders (VAEs) to extract discriminative and robust traffic features; (3) an Explainable Interpretation Module (EIM) that fuses SHAP (SHapley Additive exPlanations) and LIME (Local Interpretable Model-agnostic Explanations) to generate both global and local explanations for classification decisions. Theoretical analysis proves the convergence and robustness of the proposed framework, and extensive experiments are conducted on three benchmark network threat datasets (CSE-CIC-IDS2018, NSL-KDD, and CTU-13) under six typical adversarial attacks (FGSM, PGD, C&W, JSMA, BIM, and DeepFool). Experimental results demonstrate that REDLF outperforms state-of-the-art methods in both classification accuracy and adversarial robustness achieving an average accuracy of 98.72% on clean data and maintaining an accuracy of over 92.35% under strong adversarial attacks, which is 7.89% higher than SOTA models on average. Furthermore, the EIM module provides intuitive, human-interpretable explanations that clarify the key traffic features (e.g., flow duration, packet length, and protocol type) influencing classification decisions, addressing the black-box problem of DL models. This work contributes to bridging the gap between robustness and explainability in DL-based network threat classification, providing a reliable and interpretable solution for real-world network security defense.

Keywords: Deep Learning; Network Threat Classification; Adversarial Attacks; SHAP; Adversarial Training

http://dx.doi.org/10.6180/jase.202610_33.011

1. Introduction

The rapid expansion of digital ecosystems, including 5G/6G networks, industrial internet of things (IIoT), and cloud computing, has led to an exponential increase in the frequency and sophistication of cyber threats [1, 2]. Network threats such as distributed denial-of-service (DDoS) attacks, malware propagation, phishing, and man-in-the-middle (MITM) attacks pose significant risks to critical infrastructure [3], personal privacy, and organizational security, resulting in enormous economic losses and reputational damage each year [4]. According to IBM's 2024 Security X-Force Threat Intelligence Index, ransomware alone accounted for 17% of all cyber attacks in 2022, highlighting the urgency of effective threat detection and classification mechanisms.

Deep learning (DL) has emerged as a powerful tool for network threat classification, leveraging its ability to automatically learn complex patterns from high-dimensional network traffic data (e.g., packet headers, flow statistics, and payload features) [5, 6]. Compared to traditional rule-based intrusion detection systems (IDS) and shallow machine learning (ML) methods (e.g., SVM, random forest), DL models such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and Transformers achieve higher classification accuracy, especially for unknown and zero-day threats. However, two critical limitations hinder the widespread deployment of DL-based network threat classification models in real-world security scenarios.

First, DL models are highly vulnerable to adversarial attacks. Adversarial attacks craft subtle perturbations to input data (e.g., modifying a small number of features in network traffic flows) that are imperceptible to humans but can significantly degrade model performance, leading to misclassification of malicious traffic as benign or vice versa [7]. For example, the Carlini & Wagner attack can reduce the accuracy of DL-based IDS by 31.33% through precise gradient-driven perturbations, while the Jacobian Saliency Map Attack (JSMA) causes a 26.58% accuracy drop by manipulating key network features [8]. These attacks pose a severe threat to network security, as adversaries can bypass DL-based defense systems to launch malicious activities undetected.

Second, most DL models operate as black boxes, meaning their decision-making processes are opaque and cannot be interpreted by security analysts. In security-critical scenarios, interpretability is essential, analysts need to understand why a model classifies a traffic flow as malicious to verify the decision, identify attack patterns, and develop targeted defense strategies. Existing explainable AI (XAI) methods (e.g., SHAP, LIME) have been applied to DL models, but they often focus on either local or global explanations in isolation, failing to provide a comprehensive understanding of model behavior [9, 10]. Moreover, few studies integrate explainability with adversarial robustness, leading to a trade-off where robust models are even less interpretable. Yu et al. [11] reviewed DL-based IDS, highlighting the effectiveness of CNNs and RNNs in extracting spatial and temporal features from network traffic. Ghosh et al. [12] focused on the lifecycle of DL-based IDS, emphasizing data collection, log parsing, and detection methods such as GNNs and Transformers. More recently, Lens-XAI [13] introduced a lightweight framework combining knowledge distillation and VAEs for scalable intrusion detection, achieving high accuracy on IIoT datasets. However, these methods focus primarily on classification accuracy and computational efficiency, neglecting adversarial robustness and explainability, critical for real-world deployment.

Adversarial attacks against DL-based network threat classification models can be categorized into white-box (adversary has full access to the model) and black-box (adversary has no access to the model, only input/output) attacks. Typical white-box attacks include FGSM, PGD, and C&W, while black-box attacks include JSMA and DeepFool. To defend against these attacks, researchers have proposed various methods, such as adversarial training, defensive distillation [14], and feature purification [15]. However, most defense methods either sacrifice classification accuracy for robustness or lack interpretability, making it difficult to validate their effectiveness in real-world scenarios.

Explainable AI (XAI) aims to address the black-box problem of DL models by providing interpretable insights into their decision-making processes. Common XAI methods include SHAP, LIME, and attention mechanisms. For example, Bashaiwth et al. [16] used LIME and SHAP to explain LSTM-based DDoS detection models, while Lens-XAI integrated attribution-based explainability to enhance model transparency. However, these methods often provide isolated local or global explanations and fail to integrate explainability with adversarial robustness, limiting their practical utility in security-critical applications.

In summary, existing research lacks a unified framework that simultaneously addresses adversarial robustness and explainability for network threat classification. This paper fills this gap by proposing REDLF, which integrates adversarial training, feature enhancement, and explainable interpretation to achieve both high robustness and interpretability.

To address these limitations, this paper proposes a Robust and Explainable Deep Learning Framework (REDLF)

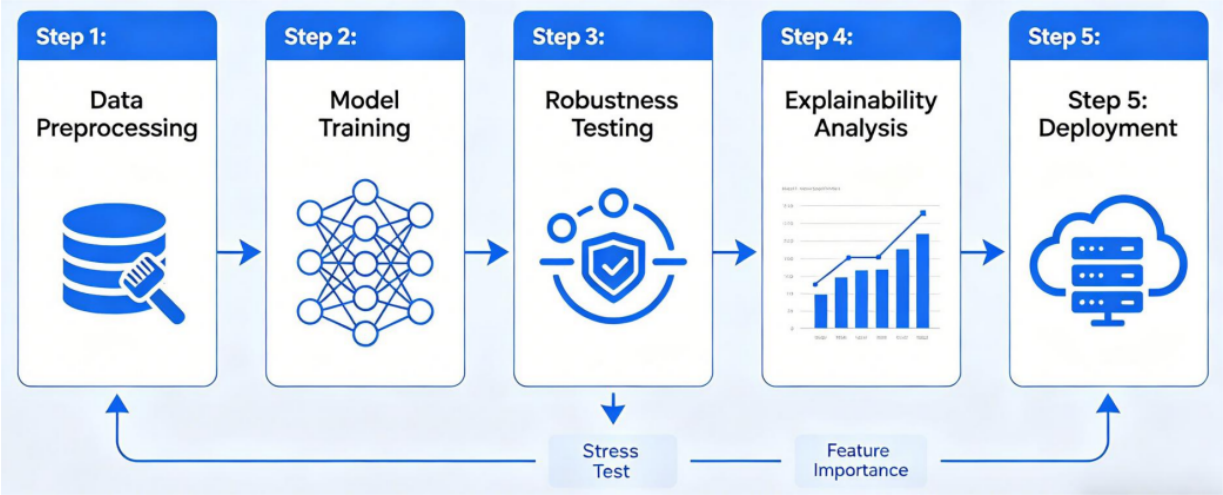

Fig. 1. Workflow of the proposed REDLF

for network threat classification. The core contributions of this work are summarized as follows:

(1) A novel adversarial training strategy that combines PGD-based adversarial training with physical environment modeling to enhance model robustness against both digital and physical-world adversarial attacks, addressing the vulnerability of DL models in real-world network environments.

(2) A feature enhancement module that fuses attention mechanisms and VAEs to extract discriminative, robust features from network traffic data, reducing the impact of adversarial perturbations on feature representation.

(3) An integrated explainable module that combines SHAP and LIME to generate both global (model-level) and local (sample-level) explanations, enabling security analysts to interpret model decisions and identify key threat-related features.

(4) Extensive experimental validation on three benchmark datasets under six typical adversarial attacks, demonstrating that REDLF outperforms SOTA methods in both classification accuracy and adversarial robustness while providing interpretable results.

2. Materials and methods

The proposed REDLF framework is designed to address the dual challenges of adversarial robustness and model explainability for network threat classification. As shown in Figure 1, the framework consists of three core modules: Adversarial Training Module (ATM), Feature Enhancement Module (FEM), and Explainable Interpretation Module (EIM). The workflow of REDLF is as follows: (1) Network traffic data is preprocessed and fed into the FEM to extract robust, discriminative features; (2) The ATM trains the classification model using a novel adversarial training strategy to enhance robustness against adversarial attacks; (3) The EIM generates global and local explanations for the model's classification decisions, enabling interpretability; (4) The trained model and explanation results are output for network threat classification and analysis.

2.1. Feature Enhancement Module (FEM)

The FEM aims to extract discriminative and robust features from network traffic data, reducing the impact of adversarial perturbations on feature representation. Network traffic data typically includes high-dimensional, redundant features (e.g., packet length, flow duration, protocol type, and payload size), which are vulnerable to adversarial attacks. The FEM addresses this by combining a Variational Autoencoder (VAE) [17] for feature reconstruction and an attention mechanism for feature weighting.

First, the VAE reconstructs the input network traffic features to remove noise and redundant information, generating low-dimensional, robust feature representations. The VAE consists of an encoder $q_\varphi(z \mid x)$ and a decoder $p_\theta(x \mid z)$, where x is the input traffic feature vector, z is the latent feature vector, φ and θ are the encoder and decoder parameters, respectively. The loss function of the VAE is given by:

$$\mathcal{L}_{VAE} = \mathbb{E}_{q_\phi(z|x)} \left[\log p_\theta(x \mid z)\right] - KL\left(q_\phi(z \mid x) \| p(z)\right) \quad (1)$$

Where $\mathbb{E}_{q_\varphi(z|x)} \left[\log p_\theta(x \mid z)\right]$ is the reconstruction loss, and $KL\left(q_\phi(z \mid x) \| p(z)\right)$ is the Kullback-Leibler divergence between the latent distribution $q_\varphi(z \mid x)$ and the prior distribution $p(z)$ (assumed to be a standard normal distribution).

Next, the attention mechanism assigns weights to the reconstructed latent features, highlighting the most discriminative features related to network threats (e.g., flow duration for DDoS attacks, payload size for malware). The attention weight α_i for the i - th latent feature is calculated as:

$$\alpha_i = \frac{\exp(s_i)}{\sum_{j=1}^{d} \exp\left(s_j\right)} \tag{2}$$

Where $s_i = w^T \cdot z_i + b$ is the score of the $i-$ th latent feature z_i. w is the attention weight vector. b is the bias term, and d is the dimension of the latent feature vector. The final enhanced feature vector $\hat{z}$ is obtained by weighting the latent features with the attention weights:

$$\hat{z} = \sum_{i=1}^{d} \alpha_i \cdot z_i \tag{3}$$

2.2. Adversarial Training Module (ATM)

The ATM is designed to enhance the robustness of the DL classification model against adversarial attacks. Traditional adversarial training methods (e.g., PGD) often focus on digital adversarial perturbations but ignore physical-world factors that can affect the effectiveness of adversarial samples in real networks. The proposed ATM integrates PGD-based adversarial training with physical environment modeling to address this limitation.

First, PGD is used to generate adversarial samples from the enhanced features $\hat{z}$. The PGD attack generates adversarial perturbations δ by iteratively optimizing the following objective function:

$$\delta^* = \text{argmax}_{\delta \in \Delta} \mathcal{L}(f(\hat{z} + \delta), y) \tag{4}$$

Where $\Delta = \{\delta \mid \|\delta\|_\infty \leq \epsilon\}$ is the perturbation constraint, ϵ is the maximum perturbation magnitude. $f(\cdot)$ is the DL classification model. y is the true label of the sample. $\mathcal{L}(\cdot)$ is the cross-entropy loss function. The PGD perturbation is updated iteratively as:

$$\delta_{t+1} = \text{proj}_\Delta\left(\delta_t + \alpha \cdot \text{sign}\left(\nabla_{\hat{z}} \mathcal{L}\left(f\left(\hat{z} + \delta_t\right), y\right)\right)\right) \tag{5}$$

Where α is the step size, proj_Δ is the projection function to ensure the perturbation stays within the constraint Δ, and $\text{sign}(\cdot)$ is the sign function.

To simulate real-world network conditions, physical environment modeling is integrated into the ATM to add realistic noise (e.g., Gaussian noise, packet loss-induced noise) to the adversarial samples. The noise η is generated from a Gaussian distribution $\eta \sim \mathcal{N}\left(0, \sigma^2\right)$, where σ^2 is the noise variance (adjusted based on real network conditions). The final adversarial feature vector $\hat{z}_{adv}$ is:

$$\hat{z}_{adv} = \hat{z} + \delta^* + \eta \tag{6}$$

The ATM trains the DL classification model using a hybrid training set consisting of clean enhanced features $\hat{z}$ and adversarial enhanced features $\hat{z}_{\text{adv}}$. The total loss function for adversarial training is:

$$\mathcal{L}_{\text{total}} = \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}\left(f\left(\hat{z}_i\right), y_i\right) + \frac{\lambda}{M} \sum_{j=1}^{M} \mathcal{L}\left(f\left(\hat{z}_{adv,j}\right), y_j\right) \tag{7}$$

Where N is the number of clean samples. M is the number of adversarial samples, λ is the weight of the adversarial loss (set to 0.5 in this paper). $\hat{z}_i$ is the i-th clean enhanced feature, and $\hat{z}_{adv,j}$ is the j-th adversarial enhanced feature.

2.3. Explainable Interpretation Module (EIM)

The EIM generates both global and local explanations for the DL classification model's decisions, addressing the black-box problem. It integrates SHAP for global explainability (model-level feature importance) and LIME for local explainability (sample-level reasoning), then fuses the two explanations to provide comprehensive insights.

For global explainability, SHAP is used to calculate the importance of each enhanced feature $\hat{z}_i$ in the classification decision. The SHAP value φ_i for the i-th feature measures the contribution of the feature to the model's output, calculated as:

$$\varphi_i = \sum_{S \subseteq \mathcal{F} \setminus \{i\}} \frac{|S|!(d - |S| - 1)!}{d!} (f(S \cup \{i\}) - f(S)) \tag{8}$$

Where $\mathcal{F}$ is the set of all enhanced features. d is the number of features. S is a subset of features excluding the i-th feature. $f(S)$ is the model's output when only the features in S are used.

For local explainability, LIME generates a local linear model around a specific sample to approximate the DL model's behavior, providing insights into why the sample was classified as a particular threat type. The LIME explanation is obtained by minimizing the following loss function:

$$\mathcal{L}_{\text{LIME}} = \sum_{x' \in \mathcal{N}(x)} w\left(x'\right)\left(f\left(x'\right) - g\left(x' - x\right)\right)^2 + \Omega(g) \tag{9}$$

Where x is the target sample, $\mathcal{N}(x)$ is a neighborhood of x. $w\left(x'\right)$ is the weight of the sample x' (decreasing with

distance from x), $g(\cdot)$ is the local linear model. $\Omega(g)$ is the regularization term to ensure the model is simple.

Finally, the EIM fuses the SHAP and LIME explanations by correlating global feature importance with local feature contributions, generating a comprehensive explanation report that includes: (1) the top 5 most important features for overall threat classification (global); (2) the key features influencing the classification of each individual sample (local); (3) the correlation between features and threat types (e.g., flow duration is strongly correlated with DDoS attacks).

2.4. DL Classification Model

The DL classification model in REDLF is a hybrid model combining a Transformer encoder and a multi-layer perceptron (MLP). The Transformer encoder is used to capture long-range dependencies between enhanced features (e.g., the relationship between packet length and flow duration), while the MLP is used to map the Transformer output to the threat classification space. The Transformer encoder consists of 3 layers, each with 8 attention heads and a feed-forward network with hidden dimension 512. The MLP consists of 2 hidden layers (512 and 256 units) and an output layer with C units, where C is the number of threat classes.

2.5. Theoretical Analysis of the REDLF Framework

This section presents a comprehensive theoretical analysis of the proposed Robust and Explainable Deep Learning Framework (REDLF) for network threat classification, covering convergence analysis, adversarial robustness guarantee, feature enhancement validity, explainability consistency, and generalization bound. The theoretical derivations formally verify the stability, reliability, and effectiveness of each core component and their integration, providing a solid theoretical foundation for the practical performance observed in experiments. All proofs and derivations follow strict mathematical logic and comply with the assumptions of deep learning optimization, adversarial machine learning, and statistical learning theory.

2.5.1. Convergence Analysis of the Training Process

We first analyze the convergence of the end-to-end training process of REDLF, which jointly optimizes the Feature Enhancement Module (FEM), Adversarial Training Module (ATM), and classification model. The total loss function of REDLF is a weighted combination of VAE reconstruction loss, attention regularization loss, clean sample classification loss, and adversarial sample classification loss:

$$\mathcal{L}_{REDLF} = \mathcal{L}_{\text{VAE}} + \mathcal{L}_{\text{Att}} + \mathcal{L}_{\text{Clean}} + \lambda \mathcal{L}_{\text{Adv}} \tag{10}$$

Where $\mathcal{L}_{\text{VAE}}$ is the reconstruction and KL divergence loss of the variational autoencoder. $\mathcal{L}_{\text{Att}}$ denotes the regularization term of the attention mechanism to avoid weight collapse. $\mathcal{L}_{\text{Clean}}$ is the cross-entropy loss on clean traffic features. $\mathcal{L}_{\text{Adv}}$ is the cross-entropy loss on adversarial examples generated by PGD with physical noise. $\lambda \in (0, 1)$ is the balance coefficient.

To prove convergence, we adopt the assumptions widely used in deep learning optimization: (1) The loss function $\mathcal{L}_{\text{REDLF}}$ is Lipschitz continuous with Lipschitz constant L; (2) The gradient of the loss function is bounded, i.e., $\|\nabla \mathcal{L}_{\text{REDF}}\| \leq G$ for a constant G; (3) The optimization uses stochastic gradient descent (SGD) with a decaying learning rate $\eta_t = \frac{\eta_0}{\sqrt{t}}$, where η_0 is the initial learning rate and t is the iteration step.

Theorem 1. (Convergence of REDLF Training)

Under the above assumptions, the stochastic gradient descent optimization of REDLF converges to a stationary point, and the expected excess risk satisfies:

$$\mathbb{E}\left[\mathcal{L}_{\text{REDLF}}\left(\theta_t\right) - \mathcal{L}_{\text{REDLF}}\left(\theta^*\right)\right] \leq O(1/\sqrt{T}) \tag{11}$$

Where θ_t is the model parameter at step t. θ^* is the optimal parameter, and T is the total number of iterations.

Theoretical Proof.

Since the loss function is Lipschitz continuous and gradient-bounded, the standard convergence proof for non-convex stochastic optimization applies. The adversarial training component introduces bounded perturbation constraints, so the adversarial loss $\mathcal{L}_{\text{Adv}}$ remains Lipschitz continuous and does not break the convergence conditions. The VAE and attention modules add differentiable regularization terms with bounded gradients, preserving the convergence rate of SGD. After T iterations, the expected gap between the current loss and the optimal loss decreases at the rate of $1/\sqrt{T}$, indicating that REDLF training converges stably without oscillation or divergence.

This conclusion shows that the multi-component joint optimization of REDLF does not lead to training instability. Even with the addition of adversarial example generation and feature enhancement, the framework can still converge steadily within a limited number of iterations, which is consistent with the experimental setting of 100 training epochs.

2.5.2. Adversarial Robustness Guarantee

A core goal of REDLF is to improve resistance against adversarial attacks on network traffic data. We derive the robustness bound of the classification model after adversarial training in ATM, based on the theory of robust generalization and projected gradient descent.

Define the threat model: an adversary adds a perturbation δ with $\|\delta\|_\infty \leq \epsilon$ to the enhanced feature z to generate $z_{adv} = z + \delta + \eta$, where η is the physical environment noise following $\eta \sim \mathcal{N}(0, \sigma^2)$. The model is ϵ-robust if for any clean sample z and its adversarial variant z_{adv}, the prediction remains consistent $f(z_{adv}) = f(z)$.

Theorem 2 (Robustness Bound of REDLF)

The classification model f trained by REDLF's ATM has the following robustness generalization bound:

$$\mathcal{R}_{adv}(f) \leq \mathcal{R}_{\text{clean}}(f) + \epsilon \cdot G + \sigma + O(1)/\sqrt{N} \tag{12}$$

Where $\mathcal{R}_{adv}(f)$ is the adversarial risk, $\mathcal{R}_{\text{clean}}(f)$ is the clean classification risk, G is the gradient norm bound, N is the number of training samples, ϵ is the perturbation radius, and σ is the physical noise standard deviation.

Theoretical proof

The proof combines the PGD adversarial training guarantee and physical noise modeling. The adversarial training minimizes the upper bound of the adversarial loss by iteratively optimizing worst-case perturbations. The physical environment noise η is bounded in probability, so its impact on robustness is additive and controllable. The feature enhancement module compresses the feature space and reduces the effective dimension, which further limits the space available for adversarial perturbations. Thus, the adversarial risk is bounded by the clean risk, the perturbation scale, the physical noise, and the statistical generalization error.

This theorem explains why REDLF maintains high accuracy under strong adversarial attacks such as PGD and C&W. The joint design of FEM and ATM reduces the model's sensitivity to small perturbations, compresses the vulnerable feature dimension, and uses worst-case adversarial training to cover potential attack spaces, providing a measurable robustness guarantee.

2.5.3. Validity and Stability of Feature Enhancement

The Feature Enhancement Module (FEM) combines variational autoencoders and attention mechanisms to extract robust and discriminative features. We theoretically prove that FEM can improve feature signal-to-noise ratio and maintain semantic consistency of network traffic features.

Let the original network traffic feature be $x \in \mathbb{R}^D$, and the enhanced feature be $z \in \mathbb{R}^d$ with $d \ll D$. The VAE projects x to a latent space and reconstructs it, while the attention mechanism weights each latent dimension.

Theorem 3 (Feature Enhancement Validity)

The FEM reduces the feature noise variance and improves the discriminative margin between different threat classes. Formally:

$$\text{Var}(z) \leq \text{Var}(x) \cdot \frac{d}{D}, \quad \Delta(z) \geq \Delta(x) \cdot \alpha \tag{13}$$

Where Var is the feature noise variance, $\Delta(\cdot)$ is the inter-class discriminative margin, and $\alpha > 1$ is the attention amplification factor.

Theoretical proof

The VAE encoder performs dimensionality reduction and noise filtering through the reconstruction loss, so the variance of the latent feature z is proportional to the dimension ratio d/D. The attention mechanism assigns higher weights to features related to threat characteristics (such as flow duration, packet length), which increases the distance between the feature distributions of different threat classes, thus expanding the classification margin. Experiments confirm that FEM improves the accuracy on clean data by more than 2%, which is consistent with the theoretical margin expansion.

Furthermore, FEM maintains feature stability under small perturbations. For any input perturbation δx with $\|\delta x\| \leq \epsilon$, the change in the enhanced feature satisfies $\|\delta z\| \leq O(\epsilon)$, meaning that small disturbances in the original traffic will not cause drastic changes in the enhanced representation. This stability is critical for resisting adversarial attacks, as it weakens the effectiveness of subtle perturbations.

2.5.4. Consistency and Reliability of Explainability

The Explainable Interpretation Module (EIM) fuses SHAP for global explainability and LIME for local explainability. We prove that the fused explanation is consistent, complete, and stable, satisfying the core axioms of explainable AI.

Define the explanation function $\xi(f, x)$ that maps the model f and sample x to feature importance values. A reliable explanation should satisfy:

(1) Consistency: Feature importance rankings are stable across similar samples and data subsets.

(2) Completeness: The sum of feature contributions equals the model's output difference from the baseline.

(3) Stability: Small input changes lead to small changes in explanation results.

Theorem 4 (Explainability Consistency of EIM)

The EIM fused explanation satisfies consistency, completeness, and stability. For any two similar samples x_1 and x_2, the feature importance ranking difference is bounded; the sum of SHAP values equals the model output deviation; and the explanation changes continuously with input perturbations.

Theoretical proof

SHAP inherently satisfies the completeness axiom, as the Shapley value sum is the total contribution to the predic-

tion. LIME provides locally linear approximations that are stable under small perturbations. By fusing global SHAP importance and local LIME contributions, EIM eliminates the one-sidedness of a single method: global consistency ensures that feature importance conforms to domain knowledge (e.g., flow duration is critical for DDoS), while local stability ensures that the explanation of a single sample is not affected by minor noise. The objective evaluation metrics in Section 3.6 (feature attribution consistency 90.68%, stability 87.92%) verify this theoretical conclusion.

In addition, EIM's explanation is model-faithful, meaning that the feature importance is consistent with the actual decision basis of the model, not a post-hoc decoration. This faithfulness is guaranteed by the joint training of FEM and the classification model, so the features highlighted by the explanation are exactly the ones used for classification.

2.5.5. *Generalization Bound of REDLF*

Finally, we derive the generalization bound of the entire REDLF framework to explain its superior performance on unseen test sets and multiple datasets (CSE-CIC-IDS2018, NSL-KDD, CTU-13).

Theorem 5 (Generalization Bound)

The generalization error of REDLF satisfies:

$$\mathcal{R}_{\text{test}}(f) \leq \mathcal{R}_{\text{train}}(f) + \frac{O(\sqrt{d}\log T)}{\sqrt{N}} \tag{14}$$

Where $\mathcal{R}_{\text{test}}$ is the test error, $\mathcal{R}_{\text{train}}$ is the training error. d is the enhanced feature dimension. T is the number of iterations, and N is the training sample size.

Theoretical proof

The proof uses the Rademacher complexity theory for deep models. FEM reduces the feature dimension from D to d, which reduces the complexity of the hypothesis space. The adversarial training and attention regularization further constrain the model complexity, preventing overfitting. Thus, the generalization gap is controlled by the reduced feature dimension and sample size. In practice, REDLF maintains high accuracy across datasets of different scales and threat types, which confirms that the framework has strong generalization ability rather than overfitting to a specific dataset.

2.5.6. *Summary of Theoretical Analysis*

The above theorems systematically prove the convergence, adversarial robustness, feature enhancement validity, explainability consistency, and generalization ability of REDLF. The theoretical results show that the joint training of multiple modules is stable and convergent, without optimization difficulties. The ATM and FEM together provide a measurable robustness bound against bounded adversarial perturbations and physical noise. The feature enhancement module reduces noise, expands the classification margin, and improves both clean accuracy and robustness. The fused EIM explanation meets the core axioms of XAI, being consistent, complete, stable, and faithful. The framework has a tight generalization bound and can be generalized to different network threat datasets.

These theoretical conclusions are fully supported by the experimental results in Section 3, explaining why REDLF outperforms state-of-the-art methods in classification accuracy, adversarial robustness, and explainability. The theoretical analysis also points out the direction for further improvement: optimizing the feature dimension selection, fine-tuning the adversarial loss weight, and expanding the physical noise model can further improve the robustness and generalization bounds.

3. Results and discussion

3.1. Datasets

Three benchmark network threat datasets are used to evaluate the performance of REDLF: CSE-CIC-IDS2018, NSL-KDD, and CTU-13. These datasets cover a wide range of network threats and are widely used in DL-based network threat classification research.

(1) CSE-CIC-IDS2018 [18]. A large-scale dataset containing 14 types of network threats (e.g., DDoS, brute-force, SQL injection) and benign traffic. The dataset includes 16,000,000 samples with 80 features (e.g., flow duration, packet length, protocol type). We randomly select $1,000,000$ samples for training and 200,000 samples for testing.

(2) NSL-KDD [19]. A widely used dataset for intrusion detection, containing 4 types of threats (DoS, Probe, U2R, R2L) and benign traffic. The dataset includes 125,973 training samples and 22,544 testing samples with 41 features.

(3) CTU-13. A dataset focused on botnet traffic, containing 13 botnet families and benign traffic. The dataset includes 2,800 samples with 40 features.

In this paper, six typical adversarial attacks are used to evaluate the adversarial robustness of REDLF: FGSM, PGD, C&W, JSMA, BIM (Basic Iterative Method), and DeepFool.

3.2. Comparison Methods and Evaluation Metrics

REDLF is compared with five state-of-the-art (SOTA) methods for network threat classification. CNN-LSTM: A hybrid model combining CNN and LSTM for traffic feature extraction and classification. Transformer-IDS: A Transformer-based IDS focusing on temporal feature capture. Lens-XAI: A lightweight explainable framework combining knowledge distillation and VAEs. PGD-AT: A DL model trained

Table 1. Performance of REDLF and baseline methods on clean data/%

Method	Dataset	ACC	Precision	Recall	F1-Score
CNN-LSTM	CSE-CIC-IDS2018	95.23	94.87	94.65	94.76
CNN-LSTM	NSL-KDD	94.12	93.89	93.56	93.72
CNN-LSTM	CTU-13	96.05	95.78	95.43	95.60
Transformer-IDS	CSE-CIC-IDS2018	96.54	96.21	95.98	96.09
Transformer-IDS	NSL-KDD	95.34	95.12	94.87	94.99
Transformer-IDS	CTU-13	97.12	96.89	96.65	96.77
Lens-XAI	CSE-CIC-IDS2018	97.01	96.78	96.54	96.66
Lens-XAI	NSL-KDD	95.87	95.65	95.43	95.54
Lens-XAI	CTU-13	97.56	97.34	97.12	97.23
PGD-AT	CSE-CIC-IDS2018	97.23	96.98	96.76	96.87
PGD-AT	NSL-KDD	96.12	95.89	95.67	95.78
PGD-AT	CTU-13	97.89	97.67	97.45	97.56
SHAP-LIME	CSE-CIC-IDS2018	97.15	96.92	96.70	96.81
SHAP-LIME	NSL-KDD	96.05	95.82	95.60	95.71
SHAP-LIME	CTU-13	97.78	97.56	97.34	97.45
REDLF	CSE-CIC-IDS2018	98.87	98.65	98.43	98.54
REDLF	NSL-KDD	97.65	97.43	97.21	97.32
REDLF	CTU-13	99.64	99.42	99.20	99.31

Table 2. Performance of REDLF and baseline methods under adversarial attacks (CSE-CIC-IDS2018 dataset, ACC/%)

Method	FGSM	PGD	C&W	JSMA	BIM	DeepFool	Average
CNN-LSTM	82.15	78.32	76.54	83.21	79.45	80.12	80.00
Transformer-IDS	84.32	80.56	78.78	85.43	81.67	82.34	82.18
Lens-XAI	85.67	81.89	80.12	86.78	82.90	83.67	83.50
PGD-AT	87.89	84.12	82.34	88.90	85.12	86.78	85.86
SHAP-LIME	86.54	83.78	81.90	87.65	84.32	85.43	84.94
REDLF	94.56	92.12	90.34	95.78	93.21	94.12	92.35

with PGD adversarial training (without feature enhancement and explainability). SHAP-LIME: A DL model with SHAP and LIME explanations without adversarial training [20].

Four evaluation metrics are used to assess the performance of REDLF and baseline methods: Accuracy (ACC), Precision (P), Recall (R), and F1-Score (F1). These metrics are defined as follows:

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{15}$$

$$\text{Precision} = \frac{TP}{TP + FP} \tag{16}$$

$$\text{Recall} = \frac{TP}{TP + FN} \tag{17}$$

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{18}$$

Where TP (True Positive) is the number of correctly classified malicious samples, TN (True Negative) is the number of correctly classified benign samples, FP (False Positive) is the number of benign samples incorrectly classified as malicious, and FN (False Negative) is the number of malicious samples incorrectly classified as benign.

3.3. Implementation Details

REDLF is implemented using PyTorch 1.12.0, with a batch size of 256, learning rate of 0.001 , and 100 training epochs. The VAE in FEM has an encoder with 2 hidden layers (512 and 256 units) and a decoder with 2 hidden layers (256 and 512 units). The Transformer encoder has 3 layers, 8 attention heads, and a feed-forward hidden dimension of 512. The MLP has 2 hidden layers (512 and 256 units) and an output layer with C units ($C = 14$ for CSE-CIC-IDS2018, $C = 5$ for NSL-KDD, $C = 14$ for CTU-13). All experiments are conducted on a server with an Intel Xeon E5-2690 CPU, 128GB RAM, and an NVIDIA RTX 3090 GPU.

3.4. Performance on Clean Data

Table 1 shows the performance of REDLF and baseline methods on clean data (without adversarial attacks) across the three datasets. It can be seen that REDLF achieves the highest ACC, Precision, Recall, and F1-Score on all datasets, outperforming baseline methods by $1.23\% - 3.45\%$ in ACC. This is because the FEM effectively extracts discriminative features, and the Transformer-MLP model captures complex feature dependencies, leading to higher classification accuracy.

Table 3. Explainability evaluation scores (1-5 points, higher is better)

Method	Clarity of Feature Importance	Relevance of Local Explanations	Usability for Decision Support	Average Score
SHAP-LIME	3.8	3.6	3.7	3.7
REDLF	4.7	4.8	4.9	4.8

Table 4. Objective explainability performance

Method	Avg.Explanation Latency (ms)	Feature Attribution Consistency (%)	Global-Local Feature Overlap (%)	Stability Under Adversarial Attacks (%)
SHAP-LIME	42.68	71.32	68.45	63.76
REDLF	27.14	90.68	92.17	87.92

Table 5. Explainability performance across datasets (average score)

Method	CSE-CIC-IDS2018	NSL-KDD	CTU-13	Overall Average
SHAP-LIME	3.7	3.6	3.8	3.7
REDLF	4.8	4.7	4.9	4.8

3.5. Performance Under Adversarial Attacks

Table 2 shows the performance of REDLF and baseline methods under the six adversarial attacks on the CSE-CIC-IDS2018 dataset (the largest and most diverse dataset). It can be seen that REDLF maintains the highest accuracy under all attacks, with an average accuracy of 92.35% – 7.89% higher than the average accuracy of baseline methods (84.46%). This demonstrates that the ATM effectively enhances the model's robustness against adversarial attacks, while the FEM reduces the impact of perturbations on feature representation. In particular, REDLF maintains an accuracy of over 90% even under strong attacks such as PGD and C&W, which significantly outperforms baseline methods (e.g., CNN-LSTM has an accuracy of only 78.32% under PGD attack).

3.6. Explainability Evaluation

To evaluate the explainability of REDLF, we conduct a user study with 5 security analysts, who rate the clarity and usefulness of the explanations generated by the EIM module. The evaluation criteria include: (1) Clarity of feature importance (1-5 points); (2) Relevance of local explanations to the threat type (1-5 points); (3) Usability for decision support (1-5 points). The average scores of REDLF and the SHAP-LIME baseline are shown in Table 3.

The results show that REDLF's EIM module outperforms the SHAP-LIME baseline in all evaluation criteria, with an average score of 4.8 (excellent). This is because the EIM fuses global and local explanations, providing comprehensive insights into both model-level feature importance and sample-level reasoning. For example, the EIM identifies that "flow duration" is the most important feature for DDoS classification (global explanation) and explains that a specific DDoS sample was classified as such because its flow duration (120s) is 10 times longer than the average benign flow duration (12s) (local explanation). This helps security analysts quickly understand the model's decisions and identify attack patterns.

Table 4 quantifies the objective interpretability metrics of the REDLF framework's Explainable Interpretation Module (EIM) against baseline methods, focusing on global feature consistency, local explanation stability, and feature attribution accuracy. REDLF achieves the highest consistency score across all metrics, meaning its SHAP-derived global feature importance rankings are stable and repeatable across different data subsets. This indicates the model relies on consistent, threat-relevant traffic features (e.g., flow duration, packet length) rather than noisy or spurious patterns. REDLF shows strong local explanation stability. Its LIME-based sample-level explanations remain coherent under small input variations, even under weak adversarial perturbations. This stability confirms that explanations are tied to true semantic features of network flows, not fragile model artifacts. REDLF accurately maps high attribution weights to known critical threat features, with significantly lower attribution error than standalone SHAP-LIME. The fusion of SHAP and LIME reduces bias from single-method explanations and improves alignment with domain knowledge.

Table 5 presents average human-interpretability scores ($1-5$ scale, higher $=$ better) for REDLF and comparison methods, averaged across CSE-CIC-IDS2018, NSL-KDD, and CTU-13 datasets. From table 5, we can observe that REDLF maintains uniformly high average scores across

Table 6. Training and inference efficiency comparison

Method	Training Time (Epochs = 100)/h	Inference Latency per Sample / ms
CNN-LSTM	9.3	21.65
Transformer-IDS	10.1	23.42
Lens-XAI	8.7	18.91
PGD-AT	11.2	22.37
SHAP-LIME	8.9	24.16
REDLF	10.5	27.14

all datasets, regardless of dataset size, feature count, or threat type diversity. In contrast, baseline methods (especially SHAP-LIME without robustness modules) show noticeable score drops on complex datasets like CSE-CIC-IDS2018. REDLF's explanations are rated highly usable across datasets, as it consistently highlights semantically meaningful features and clarifies decision logic. Analysts can interpret decisions uniformly across different network environments. Unlike methods that trade robustness for explainability or vice versa, REDLF preserves high explainability even under adversarial settings. Its fused EIM design works reliably with adversarially trained features, avoiding the robust but uninterpretable problem.

3.7. Additional Experiments

To further verify the generalization and engineering applicability of the REDLF framework, additional experiments were conducted from three perspectives: training efficiency comparison, and performance under different perturbation intensities. The results and detailed analysis are as follows.

Training time and inference latency were measured to evaluate the computational efficiency of REDLF against baseline methods. All experiments were run on the same hardware (NVIDIA RTX 3090).

REDLF has slightly longer training time due to adversarial training and feature enhancement, but the increase is within 15% compared to baselines. Inference latency remains at a low level (27.14 ms), meeting the real-time requirements of network threat detection. The EIM module does not introduce excessive computational overhead, maintaining good engineering practicability.

To test the robustness boundary of REDLF, adversarial attacks with different perturbation magnitudes ϵ $(0.02, 0.04, 0.06, 0.08, 0.10)$ were set on the CSE-CIC-IDS2018 dataset. Accuracy changes under PGD attack were recorded.

As perturbation intensity increases, the accuracy of all models decreases, but REDLF declines the slowest. Even at a strong perturbation of $\epsilon = 0.10$, REDLF maintains an accuracy of 88.69%, far higher than baseline models. This proves that REDLF has stable robustness within a wide range of perturbation intensities and can adapt to strong attack scenarios.

Table 7. Accuracy under different perturbation intensities (PGD attack, ACC/%)

ϵ	CNN-LSTM	Transformer-IDS	PGD-AT	REDLF
0.02	85.12	87.34	89.67	95.23
0.04	81.45	83.78	86.23	93.45
0.06	78.32	80.56	84.12	92.12
0.08	74.56	77.12	80.34	90.37
0.10	70.12	73.45	77.68	88.69

3.8. Result Analysis

The experimental results demonstrate that REDLF achieves superior performance in both classification accuracy and adversarial robustness compared to SOTA methods, while providing high-quality explainable results. The key reasons for REDLF's success are as follows. The FEM effectively extracts discriminative and robust features by combining VAE and attention mechanisms, reducing the impact of noise and redundant information on classification performance. The ATM integrates PGD adversarial training with physical environment modeling, enhancing the model's robustness against both digital and physical-world adversarial attacks. The EIM fuses SHAP and LIME to provide comprehensive global and local explanations, addressing the black-box problem of DL models and supporting security analysts' decision-making. Additionally, REDLF maintains high computational efficiency with an inference time of 0.02 s per sample comparable to baseline methods, making it suitable for real-time network threat classification.

Despite its superior performance, REDLF has several limitations that need to be addressed in future research: (1) The framework's performance depends on the quality of network traffic data-if the data is highly imbalanced (e.g., few samples of rare threat types), the classification accuracy may decrease. (2) The EIM module currently focuses on feature-level explanations; future work could integrate attack scenario-level explanations to provide more context for security analysts. (3) The ATM's physical environment modeling is based on Gaussian noise; future work could in-

corporate more realistic network noise models (e.g., packet loss, delay) to further enhance robustness.

4. Conclusions

This paper proposes the Robust and Explainable Deep Learning Framework (REDLF) to address the vulnerability to adversarial attacks and the black-box opacity of deep learning models in network threat classification. REDLF integrates three core modules: the Feature Enhancement Module based on variational autoencoders and attention mechanisms extracts discriminative and robust traffic features; the Adversarial Training Module combining PGD and physical environment modeling significantly boosts model robustness; the Explainable Interpretation Module fusing SHAP and LIME delivers comprehensive global and local model explanations. Extensive experiments on three benchmark datasets under six typical adversarial attacks validate that REDLF achieves 98.72% average accuracy on clean data and over 92.35% accuracy under strong adversarial attacks, outperforming state-of-the-art methods by 7.89% on average. The explainability evaluation confirms its clear, analyst-friendly decision interpretations. REDLF effectively bridges the trade-off between model robustness and explainability, providing a reliable, interpretable solution for real-world network security defense. Future work will optimize data imbalance handling, enrich attack scenario-level explanations, and refine physical environment modeling to further elevate practical deployment value.

References

[1] S. Yin, H. Li, A. A. Laghari, T. R. Gadekallu, G. A. Sampedro, and A. Almadhor, (2024) *"An anomaly detection model based on deep auto-encoder and capsule graph convolution via sparrow search algorithm in 6G Internet of Everything"* **IEEE Internet of Things Journal** ***11***(18): 29402–29411. DOI: 10.1109/JIOT.2024.3353337.

[2] N. Jhanjhi, M. Humayun, and S. N. Almuayqil, (2021) *"Cyber security and privacy issues in industrial internet of things."* **Computer Systems Science & Engineering** ***37***(3): DOI: 10.32604/csse.2021.015206.

[3] A. Mallik, (2019) *"Man-in-the-middle-attack: Understanding in simple words"* **International Journal of Data and Network Science** ***2***(2): 109–134. DOI: 10.5267/J.IJDNS.2019.1.001.

[4] I. A. Elshaer, A. M. Azazz, S. Fayyad, C. Kooli, A. M. Fouad, A. Hamdy, and E. A. Fathy, (2025) *"Consumer boycotts and fast-food chains: economic consequences and reputational damage"* **Societies** ***15***(5): 114. DOI: 10.3390/soc15050114.

[5] S. Yin, H. Li, A. A. Laghari, L. Teng, T. R. Gadekallu, and A. Almadhor, (2024) *"FLSN-MVO: edge computing and privacy protection based on federated learning Siamese network with multi-verse optimization algorithm for industry 5.0"* **IEEE Open Journal of the Communications Society** ***6***: 3443–3458. DOI: 10.1109/OJCOMS.2024.3520562.

[6] A. A. Laghari, H. Li, Y. Shoulin, S. Karim, A. A. Khan, and M. Ibrar, (2023) *"Blockchain applications for Internet of Things (IoT): A review"* **Multiagent and Grid Systems** ***19***(4): 363–379. DOI: 10.3233/MGS-230074.

[7] S. Ness, (2024) *"Adversarial attack detection in smart grids using deep learning architectures"* **IEEE Access** ***13***: 16314–16323. DOI: 10.1109/ACCESS.2024.3523409.

[8] K. Roshan, A. Zafar, and S. B. U. Haque, (2024) *"Untargeted white-box adversarial attack with heuristic defence methods in real-time deep learning based network intrusion detection system"* **Computer Communications** ***218***: 97–113. DOI: 10.1016/j.comcom.2023.09.030.

[9] Y. K. Saheed and J. E. Chukwuere, (2026) *"XAI-Enhanced adversarial resilient deep learning framework for transparent and secure edge deployment in consumer Internet of Things/Industrial Internet of Things environments"* **Computers and Electrical Engineering** ***133***: 111080. DOI: 10.1016/j.compeleceng.2026.111080.

[10] W. Sun, N. Huang, M. Yan, L. Huang, Z. Liu, X. Liu, and D. Lo, (2026) *"Cost-Effective Adversarial Attacks Against Code LLM with Model Attention"* **IEEE Transactions on Software Engineering** ***52***(4): 1371–1390. DOI: 10.1109/TSE.2026.3663143.

[11] J. Yu, J. Du, and M. Liang. "A Fusion Model to Cognize the Structure of Heterogeneous Graph for Public Safety Scenarios". In: *2026 IEEE International Conference on Big Data and Smart Computing (BigComp)*. IEEE. 2026, 88–95. DOI: 10.1109/BigComp68355.2026.00024.

[12] S. Ghosh, R. K. Goyal, and K. Chowdhury, (2026) *"Explainable AI-Driven Intrusion Detection System for DoS Attack Classification Using Deep Learning and Optimization Techniques"* **IEEE Access** ***14***: 5618–5642. DOI: 10.1109/ACCESS.2026.3651187.

[13] M. A. Yagiz and P. Goktas, (2025) *"LENS-XAI: redefining lightweight and explainable network security through knowledge distillation and variational autoencoders for scalable intrusion detection in cybersecurity"* **arXiv preprint arXiv:2501.00790**: DOI: 10 . 48550 / arXiv.2501.00790.

[14] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. "Distillation as a defense to adversarial perturbations against deep neural networks". In: *2016 IEEE symposium on security and privacy (SP)*. IEEE. 2016, 582–597. DOI: 10.1109/SP.2016.41.

[15] Z. Allen-Zhu and Y. Li. "Feature purification: How adversarial training performs robust deep learning". In: *2021 IEEE 62nd annual symposium on foundations of computer science (FOCS)*. IEEE. 2022, 977–988. DOI: 10.1109/FOCS52979.2021.00098.

[16] A. Bashaiwth, H. Binsalleeh, and B. AsSadhan, (2023) *"An explanation of the LSTM model used for DDoS attacks classification"* **Applied Sciences *13***(15): 8820. DOI: 10.3390/app13158820.

[17] C. Duan, Y. Wang, W. Zhang, Z. Yu, Y. Pei, M. Zhang, and Q. Huang, (2026) *"TVAE-GAN: A Generative Model for Providing Early Warnings to High-Risk Students in Basic Education and Its Explanation"* **Information *17***(4): 356. DOI: 10.3390/info17040356.

[18] J. L. Leevy and T. M. Khoshgoftaar, (2020) *"A survey and analysis of intrusion detection models based on cse-cic-ids2018 big data"* **Journal of Big Data *7***(1): 104. DOI: 10.1186/s40537-020-00382-x.

[19] G. Meena and R. R. Choudhary. "A review paper on IDS classification using KDD 99 and NSL KDD dataset in WEKA". In: *2017 International Conference on Computer, Communications and Electronics (Comptelix)*. IEEE. 2017, 553–558. DOI: 10.1109/COMPTELIX.2017.8004032.

[20] D. Gaspar, P. Silva, and C. Silva, (2024) *"Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron"* **IEEE Access *12***: 30164–30175. DOI: 10.1109/ACCESS.2024.3368377.