Journal of Applied Automation Technologies

Research Paper

Robust Adversarial Training for Network Traffic Classification: A ResNet-Based Approach with Protocol-Aware Perturbation

Mikołaj Kochan[1,*] **and Czesław Kamiński**[1]

[1] Faculty of Computer Science and Information Technology, Poznan University of Technology, Poznan, 60-965, Poland
*Corresponding author: mikolaj.k@put.poznan.pl

Abstract. Network traffic classification is an important component of network security, but with the development of deep learning models, concerns about adversarial attacks have also emerged. The protocol-aware adversarial training framework based on the deep ResNet architecture addresses the aforementioned issues in this study. Improve network detection accuracy and adversarial robustness. In the adversarial sample generation phase, semantic preservation constraints were introduced, and model training was conducted using system-tuned parameters. Rigorous experiments were conducted on a representative real-world dataset, and the method produced the following results: The adversarially trained classifier achieved an accuracy of 98.2% on benign traffic, maintaining over 95% accuracy and higher F1 scores under strong FGSM, PGD, and CW attacks. Compared to previous strategies, a thorough robustness index and confusion matrix analysis have shown an approximate 70% drop in adversarial accuracy. Improving robustness requires optimal regularization and deeper networks. By fine-tuning adversarial ensembles and deep residual networks, the robustness of classifiers can be significantly improved. The widespread deployment of resilient network security solutions is possible. The limitations include higher computational demands and the challenge of countering fully adaptive attacks. These issues provide a reference for future research.

Keywords: *Cybersecurity, Adversarial Learning, Deep Neural Networks, Network Traffic Classification, Robustness Analysis*

Received on 13 March 2025, Accepted on 09 August 2025, Published on 14 August 2025

Introduction

Due to the existence of networks, it is necessary to classify data to enhance network security, improve service quality, and expand new functionalities. The rapid growth of encrypted traffic, peer-to-peer applications, and evasion protocols over the past decade has increased the difficulty of reliable traffic identification [1]. The efficiency of traditional traffic classification methods is declining because the proliferation of applications and encryption protocols is hindering them [2]. By using flow-level statistics and protocol-independent features to address the aforementioned shortcomings, statistical and machine learning-based solutions have improved accuracy and flexibility [3]. Traditional models often struggle to handle new types of traffic or significant distribution changes. There is an urgent need for a classification system that can both adapt to and cope with constantly changing adversarial conditions [4]. The focus of research has shifted to robust classification models and automatic feature learning models, which can be used for complex real-world network deployments [5].

With the latest advancements in deep learning technology, automatic network traffic classification has also made significant progress. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and the more recent Residual Networks (ResNets) are typical examples of this classification technology [6]. Without the need for manual feature engineering, the model can directly extract high-level features from raw or lightly processed traffic data, thereby demonstrating good generalization ability on unseen data [7]. Deep learning classification models are sensitive to adversarial attacks, which are small-scale malicious modifications that deceive the model, despite their strong predictive capabilities [8]. Even the most advanced neural networks can

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

be manipulated by system attackers, misclassifying benign traffic or failing to recognize malicious traffic [9]. In the case of adversarial attacks, the reliability and security of deep learning models will be severely threatened. The aforementioned adversarial vulnerabilities will reduce the expected benefits of deep learning in critical task applications [10].

This paper proposes a robust network traffic classification method based on adversarial training and the ResNet architecture to address the aforementioned issues. Directly embedding adversarial sample generation into the supervised training process to enhance robustness against various adversarial attacks while maintaining the high accuracy of state-of-the-art deep learning techniques. In order to evaluate the performance of the new method under various conditions and its generalization ability, a large number of real-world traffic datasets and adversarial scenarios were used. By conducting extensive experiments and comprehensive comparisons with old models and adversarial naive baselines, the shortcomings of the old models were identified, and the new method significantly improved the robustness of automated network traffic analysis.

Related Work

Traditional Network Traffic Classification Methods

Port number matching and payload signature checking are the primary rule sets for initial network traffic classification. These methods are easy to use and understand, but many protocols have begun to use dynamic or encrypted ports, thereby hiding their original purposes, so they have become outdated [11]. Due to the aforementioned drawbacks, some researchers use statistical methods to extract flow-based features, such as byte counts, arrival interval times, and packet size distributions, to identify behavioral patterns and distinguish between different application types [12]. This approach uses feature engineering to create a set of statistically significant metrics. These metrics can be used as inputs for support vector machines, decision trees, and machine learning models [13]. Deep packet inspection can analyze protocols, but it is usually inaccurate and difficult to scale. With the widespread adoption of encryption technology, encryption has also entered everyday traffic [14].

Traditional classifiers have become increasingly difficult to scale and adapt in recent years. Manual feature selection cannot track subtle changes in traffic, especially for new encrypted or obfuscated protocols [15]. Traditional machine learning and statistics are no longer suitable for real-time applications in large-scale systems, due to the diverse and numerous types of modern network environments [16]. In order to address the constantly changing types of traffic and the evasion measures against adversarial attacks, it is recognized that there is a need to build a flexible and autonomous classification model [17].

Deep Learning Approaches in Network Traffic Analysis

With the advancement of deep learning, the field of network traffic classification is also developing rapidly. Convolutional Neural Networks (CNNs) have high accuracy and can automatically learn distinguishing features from raw byte streams or packet-level data without the need for designed features [18]. Recurrent Neural Networks (RNNs) and their gated variants, such as LSTM and GRU, enhance models that handle temporal dependencies in sequential data streams, improving streaming applications and low-latency technologies [19]. Residual networks (ResNets) were recently introduced to address the vanishing gradient problem in deep architectures. It also provides the possibility of stable multi-layer representation learning of the internal structure of complex traffic patterns [20].

Even with advancements, deep models still have their own issues. Large-scale labeled data, high computational costs, and interpretability issues still hinder its widespread use [21]. The sensitivity of deep models to out-of-distribution traffic and adversarial perturbations has garnered attention, although transfer learning and unsupervised pre-training methods have partially addressed the issue of data scarcity [22]. In practical applications, reliability decreases, and it is more likely to fail when encountering new or encrypted categories [23]. Deep learning has made significant progress in traffic classification, but security and generalization issues remain unresolved.

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

Research on Adversarial Attacks and Defenses

In cybersecurity research, there has been extensive study on the vulnerability of deep classification models to adversarial attacks. Adversarial examples are inputs that are meticulously modified from the model or training data, and it has been proven that they can lead to severe misclassifications in production systems [24]. In the field of traffic classification, the Fast Gradient Sign Method and the Projected Gradient Descent Method have been used. The accuracy of the model after rigorous training

The current defense measures are not comprehensive. For networks with high throughput and low latency, many countermeasures are too costly or only provide marginal benefits [25]. Adversarial training has been proposed to address these issues, but scalability, the ability to adapt to adversarial opponents, and the damage to the generalization of benign traffic have not been resolved. With the development of encryption and application layer evasion techniques, the demand for high-performance, large-scale, and widely applicable network traffic classifiers will continue to grow. In light of the above situation, research is being conducted to improve autonomous classification technologies, making them more resilient to advanced adversarial attacks.

Methodology

Generating Adversarial Network Traffic Samples

In order to effectively evaluate and optimize network traffic classifiers against attacks, it is necessary to develop a new technique for generating adversarial examples. This method needs to consider the discrete semantics of network flows and the inherent constraints imposed by operational protocols. In order to accommodate packet-level representation and protocol compliance, traditional continuous perturbation algorithms need to adjust the adversarial optimization process and constraint enforcement mechanisms.

A fundamental concept is to deliberately search within a controlled disturbance range to increase the likelihood of misclassification, which cannot be distinguished from real network traffic at both the syntactic and behavioral levels. This can be formulated as the solution to an adversarial optimization problem, in which the adversarial sample x^{adv} is constructed to force the classifier's decision boundary while satisfying both protocol and magnitude constraints:

$$\underset{x^{\text{adv}}}{\text{maximize}} \mathbb{I}\left[f\left(x^{\text{adv}}\right) \neq y\right] \text{ subject to } x^{\text{adv}} \in \mathcal{S}(x, \varepsilon) \quad \text{Eq.(1)}$$

In this context, $\mathcal{S}(x, \varepsilon)$ denotes a restricted neighborhood around the original feature vector x, explicitly reflecting semantic and protocol-adherence limits. The support set is typically shaped by empirical analysis of protocol fields, allowable variations in timing and sequence attributes, and operational constraints derived from real-world deployments.

To enforce valid adversarial perturbations within this set, explicit constraints are imposed on both the metric distance and the protocol legality of the manipulated sample. This dual binding framework is critical to ensuring that adversarial flows propagate through the network undetected and retain operational meaning:

$$\left\|x^{\text{adv}} - x\right\|_p \leq \varepsilon, \text{ and } \mathcal{C}\left(x^{\text{adv}}\right) = 1 \quad \text{Eq.(2)}$$

Here, $\|\cdot\|_p$ represents a norm chosen according to empirical sensitivity of the data-commonly ℓ_2 or ℓ_∞-while $\mathcal{C}(\cdot)$ is a protocol-consistency predicate that encodes grammar, state transitions, and logical field dependencies of pertinent protocols. The practical realization of $\mathcal{C}$ leverages domain-specific parsers and in-line conformance testing.

The optimization objective is tightly linked to the internal probability structure of multi-class network traffic models. It is not sufficient to merely cross the decision boundary; it is essential to minimize the classifier's confidence in the true label while maximizing the plausibility of competing labels, all under tight regularization. The crafted adversarial instance is obtained as

$$x^{\text{adv}} = \underset{z \in \mathcal{S}(x,\varepsilon)}{\text{argmin}} F_y(z) + \lambda \Omega(z, x) \quad \text{Eq.(3)}$$

where $F_y(z)$ denotes the classification confidence assigned by the model to the original class, $\Omega(z, x)$ penalizes dissimilarity from the baseline input (e.g., via Kullback-Leibler divergence or dynamic time warping for sequence-based features), and λ fine-tunes the trade-off between stealth and efficacy. The solution procedure typically

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

utilizes projected gradient descent, context-aware Jacobian computation, and semantic filtering to ensure feasible, high-impact perturbations.

Finally, rigorous adversarial evaluation demands a composite loss functional that quantifies both the model's misclassification error and the degree to which the input distribution has shifted. Such a metric is given by

$$\mathcal{L}_{\text{adv}} = \mathcal{L}\left(F\left(x^{\text{adv}}\right), y_t\right) + \alpha\mathcal{D}\left(x^{\text{adv}}, x\right) \quad \text{Eq.(4)}$$

Here, $\mathcal{L}$ represents the cross-entropy or margin-based loss relative to the adversarial target, $\mathcal{D}$ quantifies the structural divergence (potentially utilizing high-dimensional statistical geometry), and α regulates the penalty scaling associated with deviation from authentic traffic. This framework ensures the generation of high-quality adversarial instances that effectively challenge the resilience of next-generation network traffic classifiers while remaining viable within operational environments.

ResNet-Based Classifier Architecture

The ResNet-based classifier is used in this paper, aiming to more accurately identify local and global dependencies in network traffic data. By constructing the model with stacked residual units, we ensure that critical protocol and flow-level structures are preserved across layers. Each input flow is pre-processed into a multivariate feature vector maintaining timing, statistical, and payload-derived properties, and the first network layer standardizes these features to ensure stable convergence in subsequent computation.

At the core of the model, the input is successively transformed through a hierarchy of convolutional residual blocks. Unlike conventional deep neural networks where the increasing depth often leads to vanishing gradients and loss of signal fidelity, the use of identity shortcut connections in ResNet enables direct propagation of information and gradients, thereby making it possible to learn deeper, more expressive representations without degradation. The computational mechanism for a residual block is characterized by a transformation on the input, followed by an additive identity mapping, leading to an output representation that merges both the transformed and non-transformed features:

$$h_{l+1} = \sigma(\Phi(h_l; \omega_l) + h_l) \quad \text{Eq.(5)}$$

where h_l represents the feature map input at layer l, $\Phi(\cdot; \omega_l)$ encapsulates the convolutional operations with trainable parameters ω_l, and σ is a non-linear activation function. The model retains the main flow characteristics but is still able to learn the different features of various types of traffic more flexibly.

Deep within the network, the resulting activations after multiple residual blocks are aggregated via global average pooling, providing a condensed yet information-rich vector representation. For classification, this pooled representation z is mapped to the final prediction through a fully connected layer and softmax activation, yielding the predicted traffic label as:

$$\hat{y} = \text{argmax}_c G(z; \theta) \quad \text{Eq.(6)}$$

where $G(\cdot; \theta)$ denotes the terminal transformation with learned parameters θ. The architecture is systematically regularized using batch normalization at strategic points to stabilize the training dynamics and promote generalization.

To integrate multiscale representations, outputs from selected residual blocks are concatenated or averaged, giving a composite feature descriptor:

$$\xi(x) = \Psi\left(h^{(1)}(x), h^{(2)}(x), \ldots, h^{(L)}(x)\right) \quad \text{Eq.(7)}$$

where $h^{(i)}(x)$ corresponds to the output of the i-th residual block and $\Psi(\,)$ represents the chosen aggregation operator. This multiscale fusion enhances the model's robustness, particularly in the presence of adversarially perturbed flows or previously unseen traffic signatures.

Figure 1 shows the entire process from the initial preparation of the input to the stacked residual encoding and final prediction. As shown in the figure, the hierarchical convolutional modules and identity connections enhance the model's ability to express and its robustness against input disturbances.

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

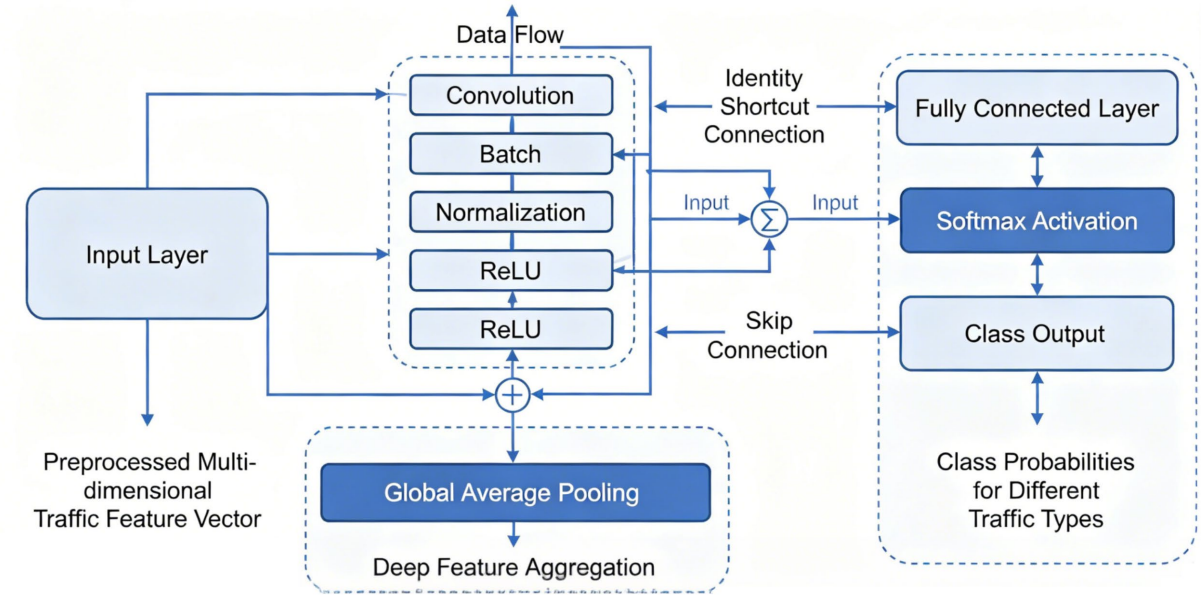

Figure 1. ResNet-based network traffic classification architecture with residual connections.

Adversarial Training Integration

By adding adversarial training to the ResNet-based classification framework, modifying the loss function, and restructuring the training process, the model is helped to learn to resist various attacks. The model receives both original samples and adversarial perturbation samples in each mini-batch iteration. This systematically places it in the most challenging environments, enabling it to learn more stable decision boundaries.

Adversarial training requires generating adversarial samples for each batch of legitimate traffic in parallel. This requires using the generation methods introduced in the previous section. These adversarial variants are dynamically generated, with smaller perturbation magnitudes and protocol-preserving constraints. The combination of benign and adversarial environment examples will affect the network's parameter updates. These two sets of data prevent the model from overfitting on the clean distribution, so it is necessary to understand the complex and significantly impactful traffic manipulation that occurs in actual attacks.

The core of this method is a universal loss function that combines the typical classification errors of clean data with the adversarial loss of the corresponding perturbed data. The overall objective for a specific mini-batch can be written as

$$\mathcal{L}_{\text{joint}} = \beta \mathcal{L}_{\text{clean}}\left(F(x), y\right) + (1-\beta)\mathcal{L}_{\text{adv}}\left(F\left(x^{\text{adv}}\right), y\right) \quad \text{Eq.(8)}$$

where $\mathcal{L}_{\text{clean}}$ and $\mathcal{L}_{\text{adv}}$ are the classification losses for clean and adversarial samples respectively, $F(\,)$ denotes the network predictions, and β is a balancing parameter modulating the influence of each component. The aforementioned design ensures that the classifier can still make accurate predictions in robust environments, thereby balancing the two during the training process.

A regularization term was added to the loss function to enhance the model's robustness against adaptive adversaries. Ensure that the model's output distribution remains stable under adversarial perturbations. In order to penalize the significant difference between the prediction distributions of clean samples and adversarial samples, this term is added to the loss function, expressed as

$$\mathcal{R}_{\text{stability}} = \gamma D\left(F(x), F\left(x^{\text{adv}}\right)\right) \quad \text{Eq.(9)}$$

where $D(\,,)$ denotes a divergence measure-such as Kullback-Leibler divergence or Total Variation distance-and γ governs the strength of regularization. By explicitly constraining the sensitivity of the model's outputs, this term acts as a safeguard against local and global shifts induced by adversarial modifications.

The resulting overall optimization objective for adversarial training is then given by the sum of these terms, ensuring comprehensive coverage across both accuracy and resilience axes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{joint}} + \mathcal{R}_{\text{stability}} \quad \text{Eq.(10)}$$

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

During the training process, the model parameters are updated through backpropagation on batches that continuously combine clean samples and adversarial samples. It is also processed through the standardized ResNet architecture. This enhanced adversarial routine makes the network more capable of generalizing to both seen and unseen perturbations. In adversarial network environments, classifiers with strong adversarial robustness characteristics can be obtained.

The adversarial training process includes data introduction, adversarial sample creation, mixed batch training implementation, and joint optimization of clean and perturbed traffic samples, as shown in Figure 2.

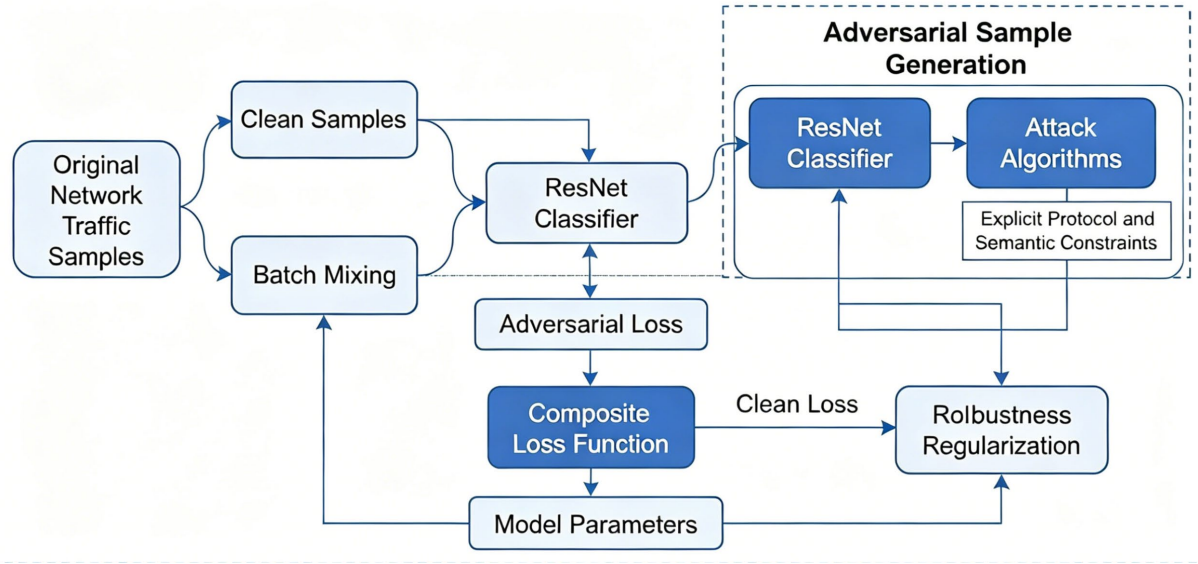

Figure 2. Adversarial training workflow for ResNet-based network traffic classification.

Experimental Setup

Dataset Description

The CICIDS2017 dataset was used for this network traffic study due to its wide recognition and rich data. By using a controlled network testbed, a dataset of a real enterprise network can be created, which includes benign traffic and various attacks such as DDoS, intrusions, and botnet behaviors. Each flow is richly annotated and includes various protocol types, with each session containing over 80 statistical and temporal features. This richness ensures that the data is sufficiently diverse under adversarial conditions to test the accuracy and generalization ability of the proposed model, and it highly represents the operational internet backbone links. To ensure the reproducibility and fairness of the experiments, all traffic sources were retained, and precise cleaning and balancing were employed to avoid sampling bias.

Evaluation Metrics

It is necessary to adopt metrics that can evaluate the general performance and adversarial attack robustness of the proposed classification method. Classification accuracy is used to represent the overall correctness level of the model in multi-class traffic classification. This is particularly effective when dealing with categories with a large number of samples. In the dataset, the proportion of correctly classified samples among all samples that were correctly classified is called accuracy. If C represents the total number of classes, $t_p^{(i)}$ the true positives for class i, and N the overall sample count, accuracy is expressed as

$$\text{Accuracy} = \frac{\sum_{i=1}^{C} t_p^{(i)}}{N} \qquad \text{Eq.(11)}$$

This metric indicates the classifier's performance across all types of traffic on a global scale. It can also quickly determine whether the model has good generalization ability, applicable to various types of input data, such as encrypted and unencrypted traffic.

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

Precision is a critical metric in the context of network security, where the cost of false alarms can be considerable. It measures the correctness of positive predictions by calculating the proportion of true positives t_p relative to all predicted positives, including both t_p and false positives f_p :

$$\text{Precision} = \frac{t_p}{t_p + f_p} \quad \text{Eq.(12)}$$

A high precision ensures that traffic flagged as belonging to a certain class (e.g., attack) is indeed likely to be correct, minimizing unnecessary response actions in practical deployments.

Recall rate is the percentage of all actual traffic events accurately identified by the model. In adversarial performance evaluation, the proportion of correctly identified positive instances to all actual positive instances is called the recall rate. The metric is as follows:

$$\text{Recall} = \frac{t_p}{t_p + f_n} \quad \text{Eq.(13)}$$

where f_n designates the number of missed true positives (false negatives). High recall is especially significant for security and anomaly detection, where undetected threats can have critical consequences.

In addition to the aforementioned performance categories, robustness metrics against adversarial attacks have also been added. Therefore, the robustness index is used to evaluate the relative decrease in accuracy under adversarial perturbations. Let A_{clean} denote the accuracy observed on unaltered (clean) traffic, and A_{adv} the accuracy achieved on adversarially manipulated inputs. The robustness index is therefore given by

$$\text{Robustness Index} = 1 - \frac{|A_{\text{clean}} - A_{\text{adv}}|}{A_{\text{clean}}} \quad \text{Eq.(14)}$$

The above formula normalizes the values. Values close to 1 indicate that the classifier has withstood the impact of adversarial attacks; conversely, lower values indicate weakness. These metrics provide a broad framework for evaluating the performance of classification models in benign environments and under adversarial attacks, and they also allow for objective comparisons through empirical experiments.

Implementation Details

The experiment will use the NVIDIA RTX 3090 GPU, and the server environment will use two 2.6GHz Xeon CPUs and 256GB of RAM. PyTorch 2.1.0 and CUDA 12.0 provide acceleration for model implementation. Dynamic feature filtering and min-max normalization are used for data preprocessing to address inter-session variability. The initial layer accepts 80-dimensional flow features, and the 20 residual blocks in the ResNet classifier all have batch normalization and ReLU activation.

The clean dataset and the adversarial dataset are divided into 30% test set and 70% training set. Adversarial samples are generated online through iterative projected gradient descent. The batch size of 128, initial learning rate of 1×10^{-3}, cosine annealing, and weight decay set to 1×10^{-5} are all key hyperparameters. According to the empirical protocol sensitivity analysis, select the perturbation boundary ε, and the flow semantics will be preserved. Train five times using different seeds, and use early stopping when the validation metrics no longer improve. The setup is reliable and statistically reasonable.

Results and Discussion

Overall Performance and Confusion Matrix Analysis

Conducting comprehensive classification evaluations in clean and adversarial environments is a method to study the per-class variations in model behavior. As shown in Figure 3(a), on the unperturbed data, the confusion matrix has a clear diagonal, indicating that most traffic types are identified very accurately, with few errors occurring, and that secondary attack types are mainly influenced by overlapping features. Figure 3(b) shows the changes in error patterns under FGSM perturbations. The botnet and infiltration flow categories are now the focus of misclassifications, indicating that the model is more sensitive to adversarial manipulations aligned with the gradient. As shown in Figure 3(c), the impact is greater under stronger PGD attacks. The off-diagonal elements of the security-sensitive categories are even larger, and greater confusion is found in the traffic with feature overlap between attack samples and benign samples.

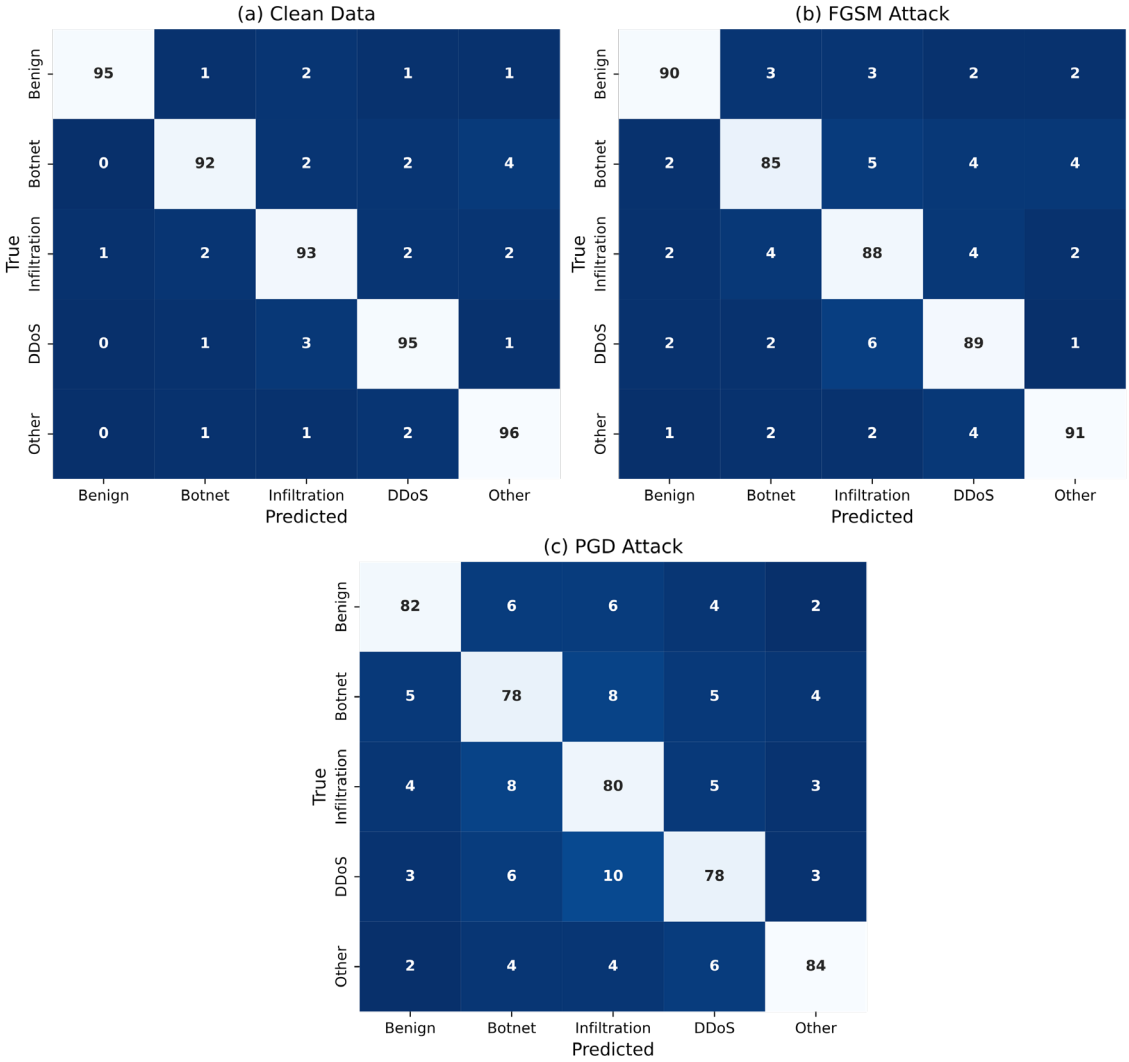

Figure 3. Confusion Matrices for Clean Data, FGSM Attack, and PGD Attack: (a)CleanData; (b)FGSMAttack; (c)PGDAttack.

According to adversarial training, under attack, these models have smaller confusion matrices and more evenly distributed classification errors. Therefore, catastrophic failures did not occur in priority detection. Quantitatively, in cases where certain categories are almost completely masked under attack, adversarial training reduces the relative drop in overall recall and macro F1 to less than 3%, while the drop for traditional models exceeds 10%. The matrix can prevent the spread of adversarial errors and protect the normal operation of the system from damage caused by malicious traffic. Many fault points in the deployed system are likely to be targeted to prevent the enemy from easily finding the objective.

Robustness under Various Adversarial Attacks

Various adversarial attacks were conducted to evaluate the robustness of the model's generalization ability. Figure 4 shows the accuracy decline trajectories and model performance of FGSM, PGD, and CW variants under different attack intensities. As shown in Figure 4(a), after the FGSM attack, the adversarially trained model still maintains a relatively high accuracy, remaining above 95% even under moderate perturbation intensity. The baseline model significantly drops to 85% under slight perturbations. Figure 4(b) shows that, in the case of PGD, this difference persists and intensifies. Adversarial training flattens the accuracy decay curve, maintains low loss in terms of performance, and remains operationally vigilant as the number of attack steps increases.

As shown in Figure 4(c), in the CW (Carlini & Wagner) attack scenario optimized for covert and effective perturbations, the accuracy of the unprotected model quickly drops below 70%, while the model trained with adversarial training maintains over 90% accuracy even in a wide perturbation environment. The reason for the

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

significant difference mentioned above is that adversarial learning can incorporate local invariance within the model's functional space. This helps to strengthen the decision boundaries, which can easily be disrupted by small adversarial attacks in the absence of adversarial training.

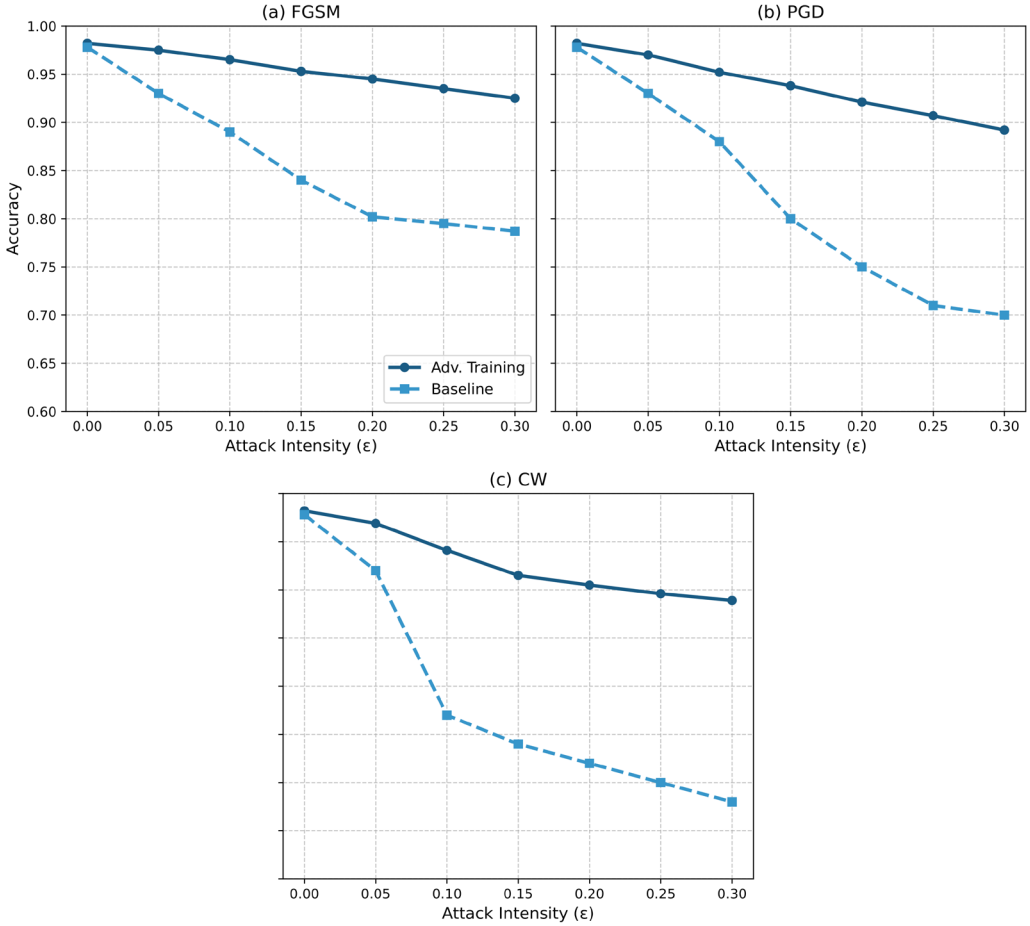

Figure 4. Accuracy Curves under Different Attacks: (a) FGSM; (b) PGD; (c) CW.

These curves also show the distribution of misclassification rates for different types of attacks; clearly, even under strong adversarial pressure, the robust model still retains its original prediction hierarchy, with the detection rates for benign and high-severity attack categories remaining much higher than those for subtle or low-impact traffic categories. The trend line in Figure 4 also indicates that the architecture of adversarial training reduces the steepness of the robustness curve deterioration while maintaining more consistent inter-class predictions across different attack intensities.

Ablation and Parameter Analysis

To determine how much of the improvement in classification accuracy and robustness is attributed to the core architecture and training parameters, systematic ablation experiments were conducted. Figure 5 shows the changes in results. Figure 5(a) shows the amount of training data. Due to the halving of the sample size, both clean and adversarial accuracies decreased by more than 6 percentage points, indicating the need for more samples to generalize and form boundaries. When the data volume approaches its maximum, the adversarial training curve shows diminishing returns; in other words, when the model capacity exceeds a certain point, further improvements are more constrained by the model capacity rather than by insufficient data volume.

As shown in Figure 5(b), the impact of network depth is significant. Increasing the number of residual blocks from 10 to 30 can enhance resistance to FGSM and PGD attacks; saturation is reached at 20 blocks, and overly deep networks may exhibit slight performance degradation due to gradient instability or overfitting. Improving

adversarial robustness is not achieved by adding a large number of layers, but by adjusting the parameters of skip connections to enhance signal transmission, thereby improving adversarial robustness.

Figure 5(c) shows the regularization strength as another axis for robustness adjustment. Dividing the penalty coefficient into two orders of magnitude, it was found that relatively small regularization (e.g., weight decay parameter of 10^{-5}) leads to the highest robustness index. Both endpoints perform poorly; the first endpoint may overfit benign structures, while the second endpoint fails to learn fine-grained category distinctions, making it very susceptible to targeted perturbations. Parameter scanning shows that the regularization bandwidth has found a certain range for adversarial robustness. By integrating rich data representations, appropriate architectural complexity, and calibrated regularization, a reliable network traffic classifier can be created. If any of the above aspects are chosen improperly, especially in adversarial environments, the system's robustness will be reduced, requiring fine-tuning to ensure stability.

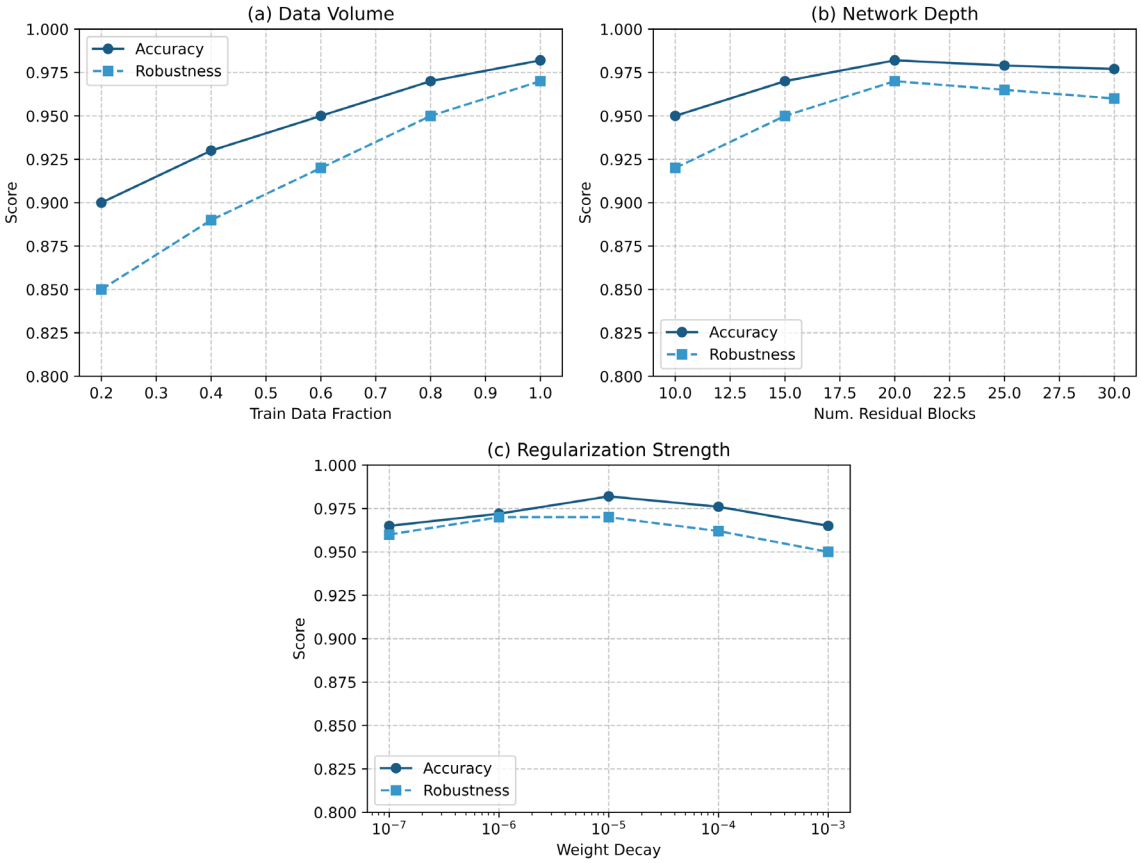

Figure 5. Ablation Study—Impact of Key Parameters: (a) Data Volume; (b) Network Depth; (c) Regularization Strength.

Confidence Scores and Real-World Implications

Predictive confidence analysis can help assess whether the system will function in practice under new risk conditions. As shown in Figure 6(a), the model's confidence score for normal traffic is close to 1, indicating that the predictions are accurate and highly reliable in the case of benign or known attacks. Automatic response and clear threat identification can be achieved. After perturbation, it becomes very different from the original state. As shown in Figure 6(b), the distribution is broader; the distribution of confidence scores is wider, and predictions with lower uncertainty are now relatively higher. This downward drift is closely related to adversarial samples that are misclassified, especially those near the class decision boundary, such as infiltrated traffic or confused botnets. High local feature space complexity has always been associated with this phenomenon.

https://doi.org/10.64972/jaat.2025v3.228p33e:447-460

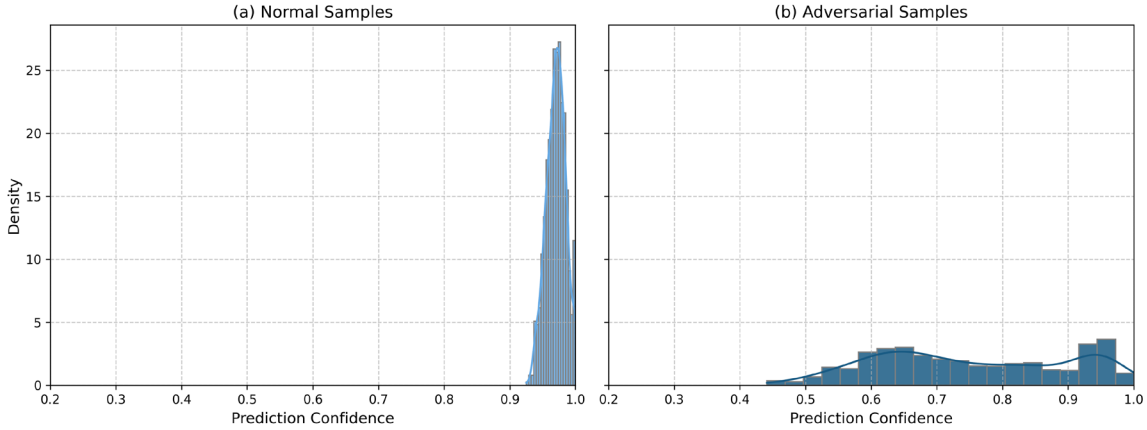

Figure 6. Distribution of Confidence Scores for Normal and Adversarial Samples: (a) Normal Samples; (b) Adversarial Samples.

The most successfully misled adversarial samples have significantly lower confidence than robust or correctly classified samples. From an operational perspective, this indicator can be used to establish a risk-sensitive threshold for the decline in prediction confidence. Traffic with low certainty will be flagged for additional inspection or manual review. Confidence calibration is a method to improve environmental awareness and enhance the model's detection accuracy in new or difficult-to-handle attack cases.

Figure 7 shows the situation of the aforementioned improvements in the context of network defense and compares different methods. As shown in Figure 7(a), under the influence of strong adaptive adversarial attacks, the proposed model maintains good robustness. Figure 7(b) shows that the classifier performs better under new traffic patterns, and Figure 7(c) shows stable performance across datasets. Immunity to adversarial drift and maintaining detection capability in new data distributions would be very useful for long-term use in real life.

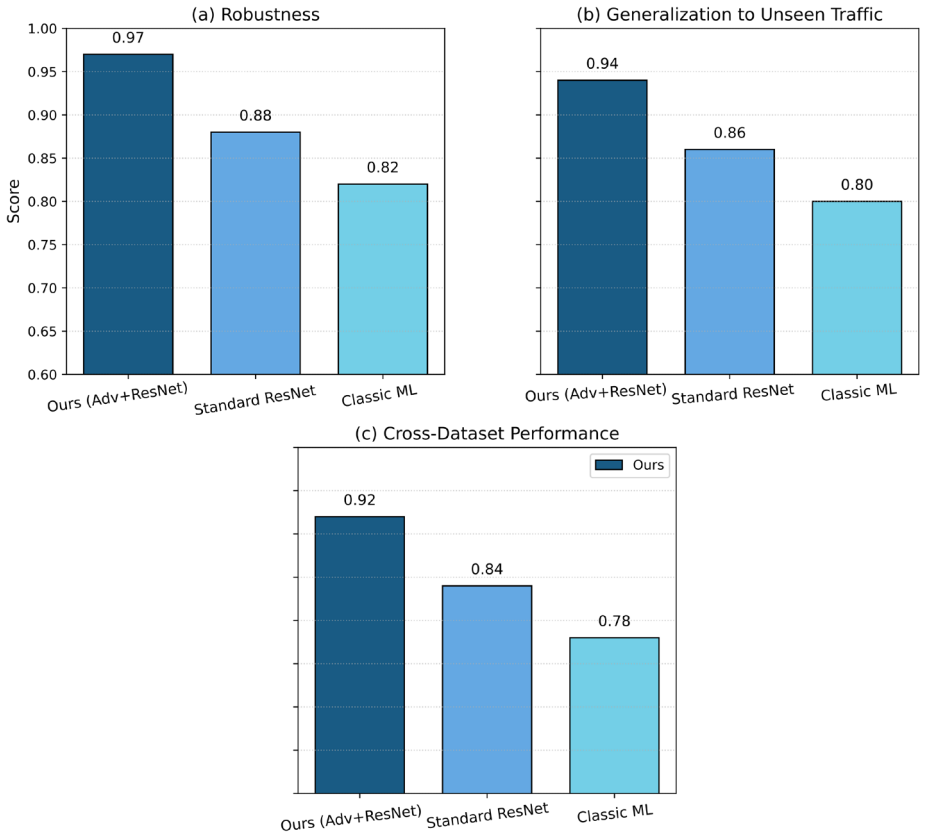

Figure 7. Robustness and Generalization Comparison with Existing Methods: (a) Robustness; (b) Generalization to Unseen Traffic; (c) Cross-Dataset Performance.

Conclusion

To address the new adversarial risks in network traffic classification, this paper introduces a robust adversarial training method based on deep residual networks and specially designed perturbation constraints. In order to systematically improve the accuracy and generality of detection, the new method includes protocols in the creation of adversarial samples and model design. A large number of experiments have shown that this method is significantly superior to previous models. Under various adversarial attacks (FGSM, PGD, and CW), it consistently exhibits higher accuracy, recall, and robustness metrics. The confusion matrix analysis also demonstrated the stability of the decision boundary and the reduction of severe misclassifications.

Through further ablation and parameter analysis, this method still performs well under various data scales, architectural complexities, and regularization settings. Determine some practical trade-offs required for large-scale deployment. The confidence distribution indicates that the model has calibrated uncertainty and supports risk-aware post-processing in the presence of unknown threats. It can be used for intelligent upgrades and trust management in real-time security environments.

Although there are still some issues. Improvements are needed to address new and subtle adversarial attacks targeting the specific invariance of the protocol. In resource-limited situations, computational cost and data requirements are also scalability issues. Future research will explore hybrid defenses that combine anomaly detection and online learning, and expand cross-domain evaluation to enhance the adaptability and robustness of these defenses in complex network environments.

Author Contributions

Mikołaj Kochan contributes to conceptualization, methodology, software, validation, analysis, investigation, data collection, draft preparation, manuscript editing, visualization, supervision. Czesław Kamiński contributes to data collection, draft preparation, manuscript editing. All authors have read and agreed with the manuscript before its submission and publication.

Funding

This research received no specific financial support from any funding agency.

Institutional Review Board Statement

Not applicable.

References

[1] Peng, Q., Fu, X., Lin, F., Zhu, X., Ning, J., & Li, F. (2025). Multi-Scale Convolutional Neural Networks optimized by elite strategy dung beetle optimization algorithm for encrypted traffic classification. Expert Systems with Applications, 264, 125729. https://doi.org/10.1016/j.eswa.2024.125729

[2] Yang, B., Arshad, M. H., & Zhao, Q. (2022). Packet-level and flow-level network intrusion detection based on reinforcement learning and adversarial training. Algorithms, 15(12), 453. https://doi.org/10.3390/a15120453

[3] Zhang, C., Costa-Perez, X., & Patras, P. (2022). Adversarial attacks against deep learning-based network intrusion detection systems and defense mechanisms. IEEE/ACM Transactions on Networking, 30(3), 1294-1311. https://doi.org/10.1109/TNET.2021.3137084

[4] Sadeghzadeh, A. M., Shiravi, S., & Jalili, R. (2021). Adversarial network traffic: Towards evaluating the robustness of deep-learning-based network traffic classification. IEEE Transactions on Network and Service Management, 18(2), 1962-1976. https://doi.org/10.1109/TNSM.2021.3052888

[5] Huang, K., Li, S., Deng, W., Yu, Z., & Ma, L. (2022). Structure inference of networked system with the synergy of deep residual network and fully connected layer network. Neural networks, 145, 288-299. https://doi.org/10.1016/j.neunet.2021.10.016

[6] Javed, H., El-Sappagh, S., & Abuhmed, T. (2024). Robustness in deep learning models for medical diagnostics: security and adversarial challenges towards robust AI applications. Artificial Intelligence Review, 58(1), 12. https://doi.org/10.1007/s10462-024-11005-9

[7] Badjie, B., Cecilio, J., & Casimiro, A. (2024). Adversarial attacks and countermeasures on image classification-based deep learning models in autonomous driving systems: A systematic review. ACM Computing Surveys, 57(1), 1-52. https://doi.org/10.1145/3691625

[8] He, K., Kim, D. D., & Asghar, M. R. (2023). Adversarial machine learning for network intrusion detection systems: A comprehensive survey. IEEE Communications Surveys & Tutorials, 25(1), 538-566. https://doi.org/10.1109/COMST.2022.3233793

[9] Khaleel, Y. L., Habeeb, M. A., Albahri, A. S., Al-Quraishi, T., Albahri, O. S., & Alamoodi, A. H. (2024). Network and cybersecurity applications of defense in adversarial attacks: A state-of-the-art using machine learning and deep learning methods. Journal of Intelligent Systems, 33(1), 20240153. https://doi.org/10.1515/jisys-2024-0153

[10] Zhou, S., Liu, C., Ye, D., Zhu, T., Zhou, W., & Yu, P. S. (2022). Adversarial attacks and defenses in deep learning: From a perspective of cybersecurity. ACM Computing Surveys, 55(8), 1-39. https://doi.org/10.1145/3547330

[11] Elmaghraby, R. T., Aziem, N. M. A., Sobh, M. A., & Bahaa-Eldin, A. M. (2024). Encrypted network traffic classification based on machine learning. Ain Shams Engineering Journal, 15(2), 102361. https://doi.org/10.1016/j.asej.2023.102361

[12] Karthika, R. A., & Maheswari, M. (2022). Detection analysis of malicious cyber attacks using machine learning algorithms. Materials Today: Proceedings, 68, 26-34. https://doi.org/10.1016/j.matpr.2022.05.510

[13] Zhang, J., Chen, Y., Ji, Q., Yu, W., Ni, L., Dai, C., ... & Luo, J. (2025). E2E-MDC: End-to-End Multi-Modal Darknet Traffic Classification with Conditional Hierarchical Mechanism. Electronics, 14(22), 4457. https://doi.org/10.3390/electronics14224457

[14] Zhai, J., Lin, P., Cui, Y., Xu, L., & Liu, M. (2023). Graphcwgan-gp: a novel data augmenting approach for imbalanced encrypted traffic classification. Computer Modeling in Engineering & Sciences, 136(2), 2069. https://doi.org/10.32604/cmes.2023.023764

[15] Ismaeel, A. G., Janardhanan, K., Sankar, M., Natarajan, Y., Mahmood, S. N., Alani, S., & Shather, A. H. (2023). Traffic pattern classification in smart cities using deep recurrent neural network. Sustainability, 15(19), 14522. https://doi.org/10.3390/su151914522

[16] Qu, A., Tang, Y., & Ma, W. (2023). Adversarial attacks on deep reinforcement learning-based traffic signal control systems with colluding vehicles. ACM Transactions on Intelligent Systems and Technology, 14(6), 1-22. https://doi.org/10.1145/3625236

[17] Chen, H. Y., & Lin, T. N. (2021). The challenge of only one flow problem for traffic classification in identity obfuscation environments. IEEE Access, 9, 84110-84121. https://doi.org/10.1109/ACCESS.2021.3087528

[18] Saffari, M., Khodayar, M., & Jalali, S. M. J. (2023). Sparse adversarial unsupervised domain adaptation with deep dictionary learning for traffic scene classification. IEEE Transactions on Emerging Topics in Computational Intelligence, 7(4), 1139-1150. https://doi.org/10.1109/TETCI.2023.3234548

[19] Wang, P., Chen, X., Shen, J., Xu, Z., Liang, F., & Du, Q. (2025). Abnormal traffic detection based on image recognition and attention-residual optimization. Frontiers in Communications and Networks, 6, 1546936. https://doi.org/10.3389/frcmn.2025.1546936

[20] Inaganti, A. C., & Sharma, V. (2022). Feature Engineering in Machine Learning for Advanced Threat Detection. Artificial Intelligence and Machine Learning Review, 3(2), 16-22. https://doi.org/10.69987/AIMLR.2022.30202

[21] Zhang, C., Wang, G., Wang, S., Zhan, D., & Yin, M. (2023). Cross-domain network attack detection enabled by heterogeneous transfer learning. Computer Networks, 227, 109692. https://doi.org/10.1016/j.comnet.2023.109692

[22] Zhang, H., Zhao, S., Liu, R., Wang, W., Hong, Y., & Hu, R. (2022). Automatic traffic anomaly detection on the road network with spatial-temporal graph neural network representation learning. Wireless Communications and Mobile Computing, 2022(1), 4222827. https://doi.org/10.1155/2022/4222827

[23] Thakkar, A., & Lohiya, R. (2023). A Review on Challenges and Future Research Directions for Machine Learning-Based Intrusion Detection System: A. Thakkar, R. Lohiya. Archives of Computational Methods in Engineering, 30(7), 4245-4269. https://doi.org/10.1007/s11831-023-09943-8

[24] Ding, Y., Zhu, G., Chen, D., Qin, X., Cao, M., & Qin, Z. (2022). Adversarial sample attack and defense method for encrypted traffic data. IEEE Transactions on Intelligent Transportation Systems, 23(10), 18024-18039. https://doi.org/10.1109/TITS.2022.3154884

[25] Qing, Y., Yin, Q., Deng, X., Zhang, X., Li, P., Liu, Z., ... & Li, Q. (2025, November). Training Robust Classifiers for Classifying Encrypted Traffic under Dynamic Network Conditions. In Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security (pp. 3564-3578). https://doi.org/10.1145/3719027.3765073