

RESEARCH ARTICLE

OPEN ACCESS

TWO-STAGE DETECT-THEN-CLASSIFY PIPELINE WITH DSAR-CNN FOR ROBUST FRAGMENT CLASSIFICATION UNDER PADDING AND BIT-FLIP ATTACKS

Teena Mary*¹ and Sreeja C S²

¹Department of Computer Science, Christ University, Bangalore, Karnataka, India – 560029.

²Center for Quantum Technologies and Complex Systems (CQTCS), Christ University, Bengaluru, Karnataka, India – 560029.

¹<https://orcid.org/0009-0007-2856-857X>^{id}, ²<https://orcid.org/0000-0002-6351-2826>^{id}

E-mail: *teena.mary@res.christuniversity.in, sreeja.cs@christuniversity.in

ARTICLE INFO

Article History

Received on: March 05, 2026

Reviewed on: April 04, 2026

Accepted on: May 01, 2026

Published on: July 31, 2026

Keywords:

Adversarial defense,
Attention mechanism,
Byte-level attacks,
Detection-based defense,
Digital forensics,
Fragment classification.

ABSTRACT

File fragment classification is fundamental to digital forensics and network security, enabling the recovery and identification of files when metadata is corrupted or absent. While state-of-the-art deep learning models on the FFT-75 benchmark achieve 65–79% accuracy on clean fragments, they exhibit vulnerability to byte-level adversarial perturbations, and evaluation across 39 byte-level attack variants shows that the baseline accuracy drops from 71.1% on clean fragments to an average of 15.8% under adversarial perturbations. This paper presents a novel two-stage detection-based defense framework tailored for file fragment classification that mitigates the accuracy-robustness trade-off commonly associated with adversarial training. The framework integrates three components: (1) an attention-based adversarial detector achieving 95.46% specificity and 91.34% recall (91.44% overall accuracy); (2) an existing baseline file fragment classifier preserving 71.1% accuracy on clean fragments; and (3) a novel Dual-Scale Attention-based Robust CNN (DSAR-CNN) specifically designed for adversarial fragment classification, achieving 68.4% accuracy on adversarially perturbed samples (evaluated on the test set of 4.6M fragments) through multi-scale depth wise separable convolution, CBAM-style dual attention, learnable position embeddings, and split global pooling. Developed and evaluated across 75 file types and 39 attack variants from the FFT-75 benchmark, the complete pipeline achieves 70.91% accuracy on clean data (0.19 percentage point penalty) and 63.84% on adversarial data, yielding 70.20% accuracy in realistic mixed-threat scenarios (90% clean, 10% adversarial) - outperforming both the undefended baseline and the standard adversarial training benchmark. The framework provides an approach for improving the robustness of forensic classification systems against byte-level adversarial attacks such as padding and bit-flipping.



Copyright ©2026 by authors and Galileo Institute of Technology and Education of the Amazon (ITEGAM). This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

I. INTRODUCTION

File fragment classification (FFC) is a primary task in digital forensics and cybersecurity that allows for the recovery and reconstruction of fragmented files when their metadata is missing or corrupted. In different application scenarios, such as cybercrime investigation, data breach, and anti-forensic operation, the accurate classification of file fragments plays an important role in evidence recovery and attribution [1]. Conventional file carving and FFC methods have evolved from rule-based heuristics to state-of-the-art machine learning and deep learning models, with recent state-of-the-art systems that achieve classification accuracies in the range of 65–79% on benchmark datasets such as FFT-75, depending on fragment size [2]. These advances have been achieved due to the adoption of CNNs, and ensemble learning, which outperformed the previous handcrafted feature-based methods by a large margin. The growing reliance on FFC in security-critical environments underlines the relevance of this task.

Indeed, accurate fragment classification not only accelerates forensic investigations but also in-creases the reliability of evidence recovery when dealing with sophisticated anti-forensic techniques and large-scale data fragmentation [3]. Despite these advances, the deployment of machine learning in FFC inherits vulnerabilities of such models to adversarial manipulation - a phenomenon well-documented in other security domains like malware detection and image classification [4] [5]. Adversarial attacks, especially those operating at the byte level, can make subtle modifications to file fragments that aim to evade detection or mislead classifiers, therefore posing a direct threat to the integrity of digital forensic processes [6]. For this reason, the development of robust and attack-resilient classification systems is growing increasingly important for both researchers and practitioners in digital forensics and cybersecurity. While FFC systems have achieved remarkable accuracy under clean conditions, they are inherently vulnerable to adversarial attacks. Empirical evaluation shows that state-of-the-art FFC models, which operate at an accuracy of 71.1% on clean data, result in catastrophic performance degradation when subjected to adversarial byte-level perturbations; the accuracy plummets as low as 15.8%, which reflects a 55.3 pp drop in the classification performance of the model [7]. These attacks exploit the discrete and high-dimensional nature of file fragments, making minimal, and often imperceptible, changes adequate for evasion or misclassification [8]. The threat is not merely theoretical: adversarial attacks have been shown to be effective in both white-box and black-box settings. In other words, the lack of adversarial defenses in current FFC systems creates a serious security loophole. This contrasts with other areas, such as malware detection and image classification, where adversarial robustness has become a central focus of research.

FFC has been slow to adopt the defense mechanisms developed and fine-tuned against the peculiarities of this specific domain [2]. Given the operational setting of FFC, where forensic investigations, court cases, and incident response rest on the accuracy of fragment classification, this is particularly concerning. A single adversarial attack can compromise the reliability of digital evidence, hamper investigations, and erode confidence in forensic results. The need for robust adversarially resilient systems is therefore both a technical and a practical imperative with significant urgency. The most prominent defense strategy against adversarial attacks in machine learning has been adversarial training, wherein models are exposed to adversarial examples during training to make them more robust [5-9]. While adversarial training can indeed enhance resistance to certain attack types, it is not without its significant drawbacks. Foremost of these is the accuracy-robustness trade-off: the enhancement in robustness to adversarial samples comes at the cost of degraded accuracy on clean data [10]. In the context of FFC, adversarial training incurs a clean accuracy penalty - a nontrivial reduction in performance, particularly in high-stakes forensic applications, in which every percent of accuracy may impact case outcomes [9]. Furthermore, adversarial training is often attack-specific, usually providing robustness only against the types of adversarial examples seen during training, and may fail to generalize to novel or adaptive attack strategies. The generation and incorporation of large volumes of adversarial samples further increase the computational overhead, limiting the scalability and practicality of this approach, especially in byte-level FFC tasks with very large datasets [10]. Other defense mechanisms explored within related domains, including ensemble learning, feature squeezing, and input transformation, have failed to demonstrate consistent or scalable success in FFC [11]. The persistent accuracy-robustness trade-off, in addition with the task's operational demands, calls for novel defense frameworks that can achieve high robustness without sacrificing clean accuracy or incurring prohibitive computational costs. This paper addresses an important gap in the study of adversarial robustness for file fragment classification. While adversarial attacks and defenses have been extensively studied in domains such as computer vision and malware detection, their implications for FFC remain largely unexplored. This work therefore presents a systematic investigation of adversarial vulnerabilities and defense mechanisms for byte-level fragment classification. The key contributions of this paper are:

1. **Detection-Based Adversarial Defense for FFC:** This work presents a two-stage detection–classification pipeline for file fragment classification in which an anomaly detection model first identifies potentially adversarial fragments before classification. By routing detected adversarial samples to a specialized robust classifier while allowing clean fragments to be processed by the baseline model, the framework enables improved robustness in mixed clean–adversarial environments without degrading baseline performance on genuine fragments.
2. **Dual-Scale Attention-based Robust CNN (DSAR-CNN):** A novel deep learning architecture is proposed combining multi-scale depthwise separable convolution, CBAM-style dual attention mechanisms, learnable position embeddings, and split global pooling.
3. **Minimal Clean Accuracy Penalty:** The detection-classification pipeline achieves 70.20% mixed-scenario accuracy (90% clean, 10% adversarial) with only a 0.19 pp penalty in clean accuracy compared to the baseline 71.1%.

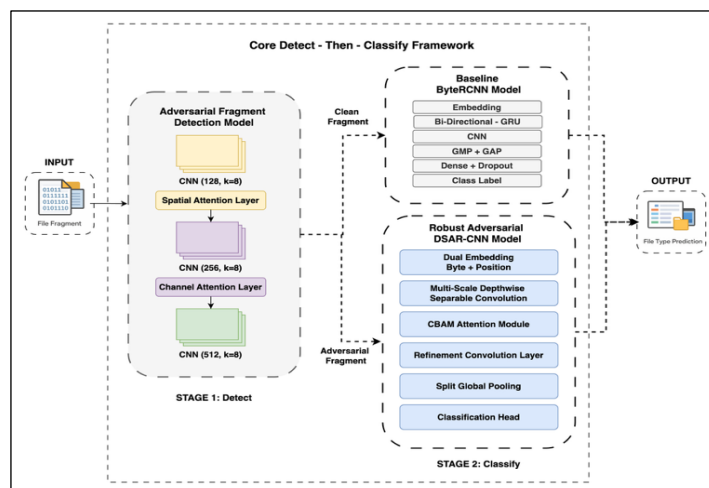


Figure 1: Two-stage detect-then-classify pipeline for adversarially robust file fragment classification.

Source: Authors, (2026).

Figure 1 illustrates the high-level architecture of the proposed two-stage defense pipeline. Each input file fragment first passes through the adversarial detection model, which performs binary classification to determine whether the fragment is clean or adversarially perturbed. Fragments identified as clean are routed to the standard file fragment classifier (ByteRCNN), preserving baseline accuracy on genuine evidence. Fragments flagged as adversarial are directed to the robust classifier (DSAR-CNN), which is specifically designed to extract discriminative features from corrupted byte sequences. This routing mechanism enables the framework to maintain high accuracy on clean data while substantially improving robustness against adversarial attacks. Together, these contributions show that anomaly detection and intelligent sample routing can achieve robustness without sacrificing either clean accuracy or operational efficiency. A comprehensive empirical comparison against a standard adversarial training baseline is provided in Section 7.2, where adversarial training is done on a balanced mix of clean and perturbed fragments across all 39 attack variants. The results confirm that the proposed pipeline outperforms adversarial training in both clean and mixed-threat scenarios. The remainder of this paper is organized as follows. Section 2 reviews existing work on file fragment classification and adversarial attacks in cybersecurity. Section 3 describes the types of attacks used in this research, including padding attacks and bit-flip attacks on file fragments. Section 4 demonstrates how these attacks reduce the baseline classifier accuracy from 71.1% to 15.8%, showing why defense is necessary. Section 5 introduces the adversarial fragment detection model. Section 6 presents the robust file fragment classifier DSAR-CNN and its architecture. Section 7 presents experimental results, showing how the defense framework improves robustness. Section 8 discusses the practical implications and limitations of the work. Section 9 concludes with a summary of key contributions to digital forensics and adversarial machine learning.

II. RELATED WORK

File fragment classification has progressed from statistical methods to sophisticated deep learning architectures. Early work by McDaniel and Heydari demonstrated that byte frequency distributions could effectively distinguish file types without relying on metadata [12]. The field advanced substantially with the introduction of the FFT-75 benchmark and the FiFTy model by Mittal et al., which achieved 77.5% average accuracy on 4096-byte fragments across 75 file types using trainable byte embeddings [13]. Subsequent research has pursued efficiency improvements: Ghaleb et al. developed lightweight CNNs with depthwise separable convolutions reaching 79% accuracy with only 100K parameters for 4096-byte fragments, while Skračić et al. proposed ByteRCNN, a hybrid recurrent-convolutional approach achieving 71.1% on the same benchmark [7-14]. Wang et al. introduced JSANet, a joint self-attention network that captures intra-sector byte-wise features and inter-sector contextual relationships, demonstrating that attention mechanisms can significantly improve fragment classification accuracy across multiple benchmarks [15]. Penrose et al. addressed the challenging problem of classifying high-entropy file fragments-including compressed and encrypted data-by proposing features based on randomness measures and discrete Fourier transform coefficients, achieving classification rates exceeding 90% for encrypted fragments [16]. More recently, Google's Magika system demonstrated that AI-powered file-type identification using optimized deep learning models can outperform traditional signature-based tools by approximately 20%, achieving precise identification within milliseconds even on CPU [17]. However, existing file fragment classifiers generally assume clean, unperturbed inputs with no mechanisms to detect or defend against adversarial manipulation, leaving forensic systems fundamentally vulnerable to byte-level attacks.

Adversarial attacks have been extensively studied in computer vision since Good-fellow et al. introduced FGSM, but their application to byte-level classification remains relatively underexplored [18]. In the malware detection domain, Kolosnjaji et al. demonstrated that gradient-based attacks against CNN architectures trained on raw executable bytes achieve high evasion rates by modifying less than 1% of bytes [19]. Demetrio et al. showed that functionality-preserving perturbations injecting benign content into executable sections severely degrade neural network detector performance [6]. Lucas et al. revealed that attack success rates can reach 90% against raw-binary classifiers without proper defenses [20]. Even more subtle bit-level manipulations can severely degrade neural network performance. For example, Kreuk et al. demonstrated that injecting a small sequence of adversarial bytes into binary executables is sufficient to cause end-to-end convolutional malware detectors to misclassify malicious samples as benign, with the adversarial payload remaining transferable across different file positions and file families [21]. Beyond input-level attacks, adversarial bit manipulations have also been studied at the model parameter level. Kummer et al. extended bit flip attacks to graph neural networks, proposing the Injectivity Bit Flip Attack that exploits architectural properties specific to message passing networks, demonstrating that targeting learnable neighborhood aggregation functions can degrade model expressivity with minimal bit modifications [22]. In the Android malware domain, Taheri et al. investigated deep learning classifier vulnerabilities against differential privacy-based adversarial attacks, demonstrating that attackers can exploit noise injection techniques to reduce classification accuracy by up to 50% [23].

These findings suggest that bit-level perturbations can significantly affect neural network predictions, a hypothesis confirmed by empirical evaluation showing accuracy degradation from 71.1% to 15.8% under padding and bit-flip attack variants. Defenses against adversarial attacks generally employ either adversarial training or detection-based methods. Zhang et al. formally characterized the accuracy-robustness tradeoff, showing that adversarial training often causes some reduction in clean accuracy without guaranteeing robustness against novel attacks [24]. Tsipras et al. further demonstrated that robustness may fundamentally conflict with standard accuracy objectives [25]. Detection-based approaches offer an alternative: Meng and Chen proposed MagNet, which identifies adversarial inputs before classification using autoencoder reconstruction, thereby avoiding accuracy penalties on clean data [26]. Goodge et al. studied autoencoder vulnerabilities in anomaly detection settings and proposed the Approximate Projection Autoencoder (APAE), demonstrating that incorporating approximate projection with feature weighting can improve robustness against adversarial attacks while maintaining detection performance on clean data [27]. Xie et al. introduced feature denoising networks that incorporate non-local means and other filtering operations within convolutional architectures, achieving state-of-the-art adversarial robustness by suppressing feature-level perturbations when combined with adversarial training [28]. Existing detection frameworks, however, are domain-specific: image-based detectors exploit spatial coherence, malware detectors leverage executable-specific structural features such as PE headers and API call sequences. File fragments span 75 heterogeneous types with format-agnostic byte patterns lacking consistent structural properties, creating a fundamental mismatch with existing defenses and necessitating a detection approach specifically tailored to byte-level file fragment analysis.

Attention mechanisms have proven effective for improving neural network robustness by dynamically focusing on discriminative features while suppressing noisy signals. Hu et al. introduced Squeeze-and-Excitation networks, which adaptively recalibrate channel-wise feature importance through learned weights [29]. Woo et al. extended this with CBAM, combining channel attention with spatial attention to identify both what features and where in the input to focus [30]. Agrawal et al. conducted a comparative study on the impact of attention mechanisms on adversarial robustness, demonstrating that CBAM-enhanced ResNet models exhibit improved resilience against PGD attacks on complex datasets like CIFAR-100, suggesting that learned salient features provide inherent defensive properties [31]. Zoran et al. proposed sequential top-down attention models for image classification that achieve state-of-the-art adversarial robustness by processing inputs through multiple attention steps, revealing that increasing the number of attention glances creates a computational race between attacker and defender while producing adversarial examples with global, perceptible structures [32]. In malware detection, Al-Ghanem et al. achieved 99.5% accuracy with MADANET by learning to emphasize discriminative byte patterns, while Yakura and Sakuma used attention to extract important byte sequences characterizing malware functionality [33] [34]. Despite these successes, attention mechanisms remain underexplored for addressing adversarial robustness in file fragment classification. The application of dual attention mechanisms - combining channel and spatial attention in a hierarchical structure enables synergistic improvements where the combined effect exceeds the sum of individual contributions, providing a promising direction for robust fragment classification under adversarial conditions.

III. THREAT MODEL AND ATTACK TAXONOMY

The threat model and attack taxonomy presented here are built upon the comprehensive adversarial evaluation framework established in prior work, extended here to evaluate defense mechanisms. The threat model assumes a non-adaptive adversary who has no query access to the classifier architecture or training data. The attacks are pre-specified input-constrained transformations rather than optimized per-sample adversarial examples. The attacker will aim to induce misclassification through byte-level perturbations while maintaining the fragment size exactly at 512 bytes. This restriction follows realistic forensic scenarios in which fragments are recovered from storage media stripped of metadata and context. The attacks considered are format-agnostic, based on general byte-level manipulations rather than exploiting file-type-specific structural properties. Two primary attack categories represent realistic adversarial perturbations: padding attacks and bit-flip attacks. Padding attacks overwrite bytes near the fragment tail using three mechanisms: null-byte padding (adding low-entropy zeros), random-byte padding (adding random bytes), and cross-file padding (adding authentic bytes from different file types). For each cross-file padded fragment, the source file type is selected uniformly at random from the remaining 74 file types, and the replacement bytes are drawn from a randomly sampled fragment of that type. This ensures diverse cross-type contamination patterns and prevents the detector from memorizing specific source-target type associations. The fragment length remains fixed at 512 bytes throughout all attacks. Each padding type is tested at five sizes-16, 32, 64, 128, and 256 bytes-producing 15 attack variants. Bit-flip attacks corrupt individual bytes in two ways: random bit-flipping modifies bytes distributed throughout the fragment, while header-targeted bit-flipping concentrates modifications in the first 64–128 bytes where file signatures reside. These header-targeted attacks remain format-agnostic in that they do not parse format-specific structures (e.g., PDF dictionaries or JPEG markers) but rather exploit the generic concentration of discriminative magic bytes in early file positions. With four corruption rates and three bit-flip intensities, bit-flip attacks yield 24 variants, resulting in a total attack taxonomy of 39 distinct strategies. These details are summarized in Table 1:

Table 1: Byte-Level Adversarial Attack Taxonomy for FFT-75 File Fragment Classification.

Attack family	Attack type	Variants (count)	Parameter grid	Intended forensic realism
Padding	Null-byte padding	5	Padding sizes $P \in \{16, 32, 64, 128, 256\}$ bytes	Models zero-filled buffers, incomplete reads, and partially overwritten sectors that naturally occur on disks.
Padding	Random-byte padding	5	Padding sizes $P \in \{16, 32, 64, 128, 256\}$ bytes	Simulates noise injection, disk corruption, and transmission errors producing high-entropy tails in fragments.
Padding	Cross-file padding	5	Padding sizes $P \in \{16, 32, 64, 128, 256\}$ bytes	Appends authentic bytes from other file types to induce file-type confusion and hybrid statistical profiles.
Bit-flip	Random bit-flip	12	$k \in \{1, 5, 10, 20\}\%$ of bytes; $n \in \{1, 2, 4\}$ bits per selected byte	Emulates random memory faults, media degradation, and low-level corruption without obvious structural artifacts.
Bit-flip	Header-targeted bit-flip	12	Same (k, n) grid as random bit-flip, restricted to header region (first 64–128 bytes)	Targets magic numbers and header metadata, reflecting adversaries corrupting type-identifying bytes while preserving length.

Source: Authors, (2026).

Unlike adversarial attacks studied in computer vision, which often aim to produce imperceptible perturbations, the objective in file fragment classification is different. In digital forensics, fragments frequently originate from partially overwritten sectors, damaged storage media, or intentionally manipulated files where substantial portions of the fragment may be corrupted.

Consequently, the attack taxonomy includes perturbations ranging from small bit flips to large-scale byte overwrites (e.g., 256-byte padding) to model both subtle corruption and severe anti-forensic manipulation scenarios. These perturbations therefore represent a spectrum of realistic byte-level corruption conditions rather than purely imperceptible adversarial examples. The goal is therefore not only to evaluate robustness against subtle adversarial manipulations, but also to assess the resilience of fragment classifiers under severe corruption conditions that may arise in real-world forensic investigations. Scope of Evaluation. The defense framework is trained and evaluated exclusively against the 39 non-adaptive attack variants defined in this taxonomy. Since both the detector and robust classifier observe all attack types during training, the evaluation measures in-distribution robustness rather than generalization to novel attack strategies. An adaptive adversary with full pipeline knowledge could potentially craft composite perturbations (e.g., simultaneous padding and bit-flipping) or target the detector's learned attention patterns directly - scenarios not evaluated here. Hence, adaptive threat evaluation is identified as a priority for future work (Section VIII.2). The evaluation uses the FFT-75 test partition as the clean fragment source, comprising 768,000 fragments distributed evenly across 75 file types at 512 bytes each (see Section IV.2 for the complete dataset partitioning strategy). Applying all 39 attack variants to each clean fragment generates 29.952 million adversarial samples, for a total dataset of 30.72 million fragments. This dataset is split into training (70%, 21,504,000 fragments), validation (15%, 4,608,000 fragments), and testing (15%, 4,608,000 fragments), with each split independently allocating 70%, 15%, and 15% of both clean and adversarial samples respectively. The training set comprises 20,966,400 adversarial samples and 537,600 clean samples, resulting in a 39:1 adversarial-to-clean class imbalance.

IV. BASELINE BYTERCNN AND UNDEFENDED VULNERABILITY ANALYSIS

The purpose of this section is to establish a quantitative baseline of adversarial vulnerability in file fragment classification systems. This baseline serves two key goals: first, it measures the security gap in current state-of-the-art FFC models; and second, it validates that the 39-variant attack taxonomy is feasible and effective in practice.

IV.1 BASELINE MODEL SELECTION AND JUSTIFICATION

Selecting the right baseline model is essential for a fair assessment. Three candidate models were evaluated based on these criteria: accuracy on FFT-75 for 512-byte fragments, availability of public code, and practical usability. Current file fragment classification models typically achieve between 65% and 79% accuracy on FFT-75, depending on the fragment size and model type. FiFTy achieves an average of 77.5% across 4096 fragment size, but for 512-byte fragments specifically, FiFTy only reaches 65.6% [13]. ByteRCNN, published in 2023, achieves 71.1% on 512-byte fragments, which is notably higher [7]. JSANet, a transformer-based model with joint self-attention mechanisms, reported performance on the FFT-75 benchmark but does not provide a specific 512-byte accuracy figure in publicly available documentation [15]. Since ByteRCNN offers public code and achieves one of the highest documented 512-byte accuracy (71.1%), it was selected as the baseline model. It represents current state-of-the-art and was published by Skračić et al. in IEEE Access 2023. ByteRCNN is a hybrid recurrent-convolutional model designed for file-fragment type identification. It incorporates a byte-embedding layer that transforms raw byte values into trainable dense representations, followed by recurrent and convolutional components that capture sequential and local patterns within byte sequences. The model is evaluated on the FFT-75 dataset, specifically Scenario 1, which contains 75 file types. Experiments include both 512-byte and 4096-byte fragments, where ByteRCNN achieves 71.1% accuracy on 512-byte samples and 83.9% accuracy on 4096-byte samples.

IV.2 TEST DATA AND EXPERIMENTAL SETUP

The FFT-75 Scenario 1 dataset comprises 7,680,000 fragments (102,400 fragments per file type across 75 file types at 512 bytes each), partitioned by the dataset creators into 80% training (6,144,000 fragments), 10% validation (768,000 fragments), and 10% test (768,000 fragments). The ByteRCNN baseline model was trained on the official FFT-75 training partition (6,144,000 fragments), consistent with the original evaluation protocol. To construct the adversarial evaluation benchmark, the 768,000-fragment test partition (10,240 fragments per file type) was used as the clean fragment source. Applying all 39 attack variants to each clean fragment generates $768,000 \times 39 = 29,952,000$ adversarial samples, for a total of 30,720,000 fragments (clean + adversarial). This combined dataset was then split into training (70%, 21,504,000 fragments), validation (15%, 4,608,000 fragments), and testing (15%, 4,608,000 fragments) using stratified sampling to preserve the clean-adversarial ratio in each split. The training partition is used exclusively for training the adversarial detection model and the DSAR-CNN robust classifier; it is not used for ByteRCNN training. Importantly, the training, validation, and test partitions are constructed using stratified sampling with strictly disjoint fragment sets - no individual fragment or its adversarial variants appear in more than one partition. This ensures that the defense models are evaluated exclusively on fragments unseen during their training, preventing data leakage. The FFT-75 test partition (768,000 fragments) was selected as the adversarial generation source rather than the larger training partition (6,144,000 fragments) to maintain computational tractability; generating all 39 attack variants from the full training set would yield over 245 million samples. Since both partitions are drawn from the same underlying file corpus with identical per-type distributions, this choice does not introduce distributional bias.

The test dataset consists of these parts:

- **Clean test fragments:** 115,200 fragments (15% of the FFT-75 test partition), distributed equally across 75 file types (1,536 fragments per type).
- **Adversarial test fragments:** Generated by applying each of the 39 attack variants to the same 115,200 clean fragments. This produces $115,200 \times 39 = 4,492,800$ adversarial samples. Each attack is applied independently; no combined or cumulative attacks are tested.
- **Total test dataset:** 4,608,000 fragments (115,200 clean + 4,492,800 adversarial).

This structure ensures that all attacks are tested on identical baseline data, making it easy to compare how different attack types affect the classifier.

For padding attacks: each 512-byte fragment has its last N bytes (where $N \in \{16, 32, 64, 128, 256\}$) removed and replaced with corrupted bytes using one of three methods: null bytes (0x00), random bytes (0x00–0xFF), or authentic bytes from a different file type. The result is always 512 bytes long, with the first $(512 - N)$ bytes intact and the last N bytes corrupted. For bit-flip attacks: either a random selection of $k\%$ of bytes ($k \in \{1, 5, 10, 20\}$) or just the header region (first 64–128 bytes) is chosen, then n random bits per selected byte ($n \in \{1, 2, 4\}$) are flipped. This generates 12 random variants and 12 header-targeted variants. Multiple metrics measure vulnerability from different angles:

1. **Absolute accuracy** is the fraction of correctly classified fragments, as given in the formula:

$$\text{Acc} = \frac{\text{Correct predictions}}{\text{Total samples}} \quad (1)$$

2. **Accuracy drop** (percentage points) shows raw performance loss, as given in the formula:

$$\Delta_{\text{acc}} = \text{Acc}_{\text{clean}} - \text{Acc}_{\text{adv}} \quad (2)$$

3. **Relative vulnerability** shows the percentage of capability lost, as given in the formula:

$$V_{\text{rel}} = \frac{\text{Acc}_{\text{clean}} - \text{Acc}_{\text{adv}}}{\text{Acc}_{\text{clean}}} \times 100\% \quad (3)$$

Comparison to uniform class probability: For 75 file types, accuracy of randomly choosing the correct file type is $1/75 \approx 1.33\%$.

IV.3 CLEAN BASELINE PERFORMANCE

Before evaluating adversarial robustness, the clean performance of the baseline model was verified. The ByteRCNN model was trained from scratch on the FFT-75 training partition (6,144,000 fragments) using the publicly available architecture and training configuration, as pre-trained weights are not provided by the original authors. When evaluated on 115,200 clean test fragments (held out from the FFT-75 test partition, disjoint from all training data), the model correctly classified 81,907 fragments, resulting in a clean accuracy of 71.096% ($\approx 71.1\%$). This result is consistent with the performance reported in the original ByteRCNN study for 512-byte fragments, confirming that the re-implementation reproduces the expected baseline behavior.

IV.4 ADVERSARIAL VULNERABILITY RESULTS

ByteRCNN's clean accuracy on 512 size fragments is 71.1% [7]. From the empirical evaluation, it was found that across all 39 attacks, ByteRCNN's average adversarial accuracy comes to $\text{Acc}_{\text{adv}} = 15.8\%$. This represents an absolute drop of:

$$\Delta_{\text{acc}} = 71.1\% - 15.8\% = 55.3 \text{ percentage points} \quad (4)$$

In relative terms, the model retains only 22.2% of its clean performance, corresponding to a relative vulnerability of:

$$V_{\text{rel}} = \frac{55.3}{71.1} \times 100\% = 77.8\% \quad (5)$$

For a 75-class classification problem, the probability of correct classification under uniform class selection is 1.33% ($1/75$). On clean data, ByteRCNN achieves an accuracy of 71.1%, which corresponds to a performance level 53.5 times higher than the uniform classification probability ($71.1\% \div 1.33\%$). Under adversarial conditions, the accuracy decreases to 15.8%, representing only 11.9 times the uniform probability level ($15.8\% \div 1.33\%$). This comparison indicates that adversarial perturbations substantially reduce the model's effective discriminative capability, bringing its performance significantly closer to the uniform baseline. A summary of the accuracy degradation for the different attack types is given in Table 2:

Table 2: Accuracy Degradation on Baseline ByteRCNN Model for Different Attack Types.

Attack Type	Count	Lowest Acc (%)	Highest Acc (%)	Average Acc (%)	Average Drop (pp)	Relative Vulnerability (%)
Null-byte padding	5	4.1	22.4	12.3	58.8	82.7
Random-byte padding	5	3.7	21.5	11.5	59.6	83.8
Cross-file padding	5	2.7	19.2	9.8	61.3	86.2
Random bit-flip	12	6.4	38.8	20.2	50.9	71.6
Header bit-flip	12	5	34.7	17.1	54	75.9
Overall	39	2.7	38.8	15.8	55.3	77.8

Source: Authors, (2026).

Padding attacks (11.2% average) are significantly more damaging than bit-flip attacks (18.7% average). The 7.5 pp difference occurs because padding overwrite or replace existing bytes from the fragment, destroying information entirely. Bit-flip attacks only change specific byte values while keeping the file structure mostly intact. Cross-file padding is the hardest attack (9.8% average). Unlike null-byte or random-byte padding which introduces obvious anomalies, cross-file padding adds real byte sequences from other file types. This actively confuses the classifier by introducing correct looking but contextually wrong file signatures. A consistent pattern emerges across all attack types: performance degradation scales with perturbation magnitude. Table 3 ranks the attack severity (top 5) from the most to the least damaging for the model based on model accuracy.

Table 3: Attack Severity Ranking: Most to Least Damaging Variants by Residual Accuracy.

Rank	Attack Type	Parameters	Residual Accuracy (%)	Accuracy Drop (pp)
1	Cross-file padding	256 B	2.7	68.4
2	Random-byte padding	256 B	3.7	67.4
3	Null-byte padding	256 B	4.1	67
4	Cross-file padding	128 B	5	66.1
5	Header bit-flip	k=20%, n=4	5	66.1
-	-	-	-	-
35	Random bit-flip	k=5%, n=1	28.3	42.8
36	Header bit-flip	k=1%, n=2	28.8	42.3
37	Random bit-flip	k=1%, n=2	32.9	38.2
38	Header bit-flip	k=1%, n=1	34.7	36.4
39	Random bit-flip	k=1%, n=1	38.8	32.3

Source: Authors, (2026).

For padding attacks, null-byte padding accuracy decreases from 22.4% with 16-byte perturbations to 4.1% with 256-byte perturbations. Random-byte padding shows a similar trend, dropping from 21.5% to 3.7% as padding size increases. Cross-file padding demonstrates the steepest degradation, falling from 19.2% to 2.7% across the same range. Bit-flip attacks follow the same pattern, though with higher residual accuracy: random bit-flip accuracy declines from 38.8% (k=1%, n=1) to 6.4% (k=20%, n=4), while header-targeted bit-flip drops from 34.7% to 5.0% under equivalent perturbation intensity. This monotonic relationship reflects a fundamental principle: as perturbations grow larger, the proportion of authentic, discriminative bytes decreases, leaving fewer patterns for the classifier to exploit. Consequently, accuracy degrades proportionally with the volume of corrupted information. The least damaging attack (random bit-flip with only 1% of bytes affected and 1 bit flipped per byte) still reduces accuracy from 71.1% to 38.8% - a 32.3 pp drop. This shows clearly that adversarial vulnerability is not limited to severe attacks; even subtle modifications cause performance loss. No single attack variant preserves reasonable performance; the highest adversarial accuracy across all 39 variants is 38.8%, meaning the lowest drop is 32.3 pp.

IV.4.1 Statistical Confidence

With 4,492,800 adversarial test samples (n), the measurements are precise. The 95% confidence interval for the 15.8% adversarial accuracy (p) is computed using the standard error for a binomial proportion:

$$SE = \sqrt{\frac{p(1-p)}{n}} = \sqrt{\frac{0.158 \times 0.842}{4,492,800}} = 0.000172 \quad (6)$$

The binomial proportion method was chosen because classification accuracy represents binary outcomes (correct or incorrect), and with 4.5 million test samples, the Central Limit Theorem implies that the sampling distribution of the estimated proportion is well approximated by a normal distribution, making this a standard statistical approach for reporting accuracy confidence intervals in machine learning research. The 95% confidence interval is therefore calculated as:

$$CI_{95\%} = p \pm z_{\alpha/2} \times SE = 0.158 \pm 1.96 \times 0.000172 = [15.77\%, 15.83\%] \quad (7)$$

This ± 0.034 pp range demonstrates that the 15.8% figure is measured with high precision. The large sample size ensures that small variations in true accuracy are captured with negligible measurement error.

IV.5 UNDERSTANDING THE VULNERABILITY

Four factors explain why adversarial performance is poor:

- 1. Statistical patterns are disrupted.** ByteRCNN learns byte-level statistics like n-gram frequencies and entropy profiles that are unique to authentic files. Padding directly destroys these patterns by replacing authentic bytes with foreign or random data. Bit-flip attacks corrupt the exact byte values that contribute to these statistics.
- 2. Position matters.** The model learns that file headers (positions 0–128) contain distinctive signatures. Header-targeted bit-flip attacks (17.1% average) are more damaging than random bit-flip attacks (20.2% average) because they deliberately corrupt these critical positions.
- 3. Authentic forgeries confuse the model.** Cross-file padding is especially effective (9.8% average) because it injects real byte sequences from other file types. The classifier encounters statistically valid but contextually incorrect patterns, causing systematic misclassification rather than simple information loss.
- 4. The model has never seen adversarial examples.** ByteRCNN was trained only on clean data. It has no learned mechanisms for distinguishing authentic from corrupted regions. Robustness must be explicitly designed.

The empirical findings establish that file fragment classifiers cannot withstand byte-level attacks without explicit defense mechanisms. Simple, easy-to-implement attacks - requiring no sophisticated optimization or targeted knowledge - cause complete classification failure.

Adversaries need not develop complex, model-specific attacks; format-agnostic byte-level perturbations suffice to entirely compromise system integrity. This reality makes defense mechanisms mandatory for any deployment of these systems in security-critical applications. Based on this baseline vulnerability analysis, effective defense mechanisms should satisfy three concrete performance targets.

1. First, adversarial accuracy should exceed 60% to recover most of the capability lost under attack (compared to the current 15.8%).
2. Second, clean accuracy should remain at or above 70% to keep the penalty on genuine evidence minimal.
3. Third, performance on realistic mixed-threat scenarios - where 90% of fragments are clean and 10% are adversarially perturbed [35] - should exceed 68%. The baseline mixed accuracy is computed as follows:

$$\text{Acc}_{\text{mixed}} = 0.9 \times 71.1\% + 0.1 \times 15.8\% = 65.57\% \quad (8)$$

Any defense achieving these three targets simultaneously will represent an advance in adversarial robustness for file fragment classification.

V. ADVERSARIAL FRAGMENT DETECTION MODEL

The first stage of the defense pipeline is an adversarial detection model that acts as a gatekeeper in front of the file fragment classifiers. Its role is to decide, for each 512-byte fragment, whether the fragment is clean or adversarially perturbed and to route it accordingly. Fragments predicted as clean are passed to the baseline ByteRCNN classifier, while fragments predicted as adversarial are forwarded to the robust classifier introduced in the next section. This design allows the standard FFC model to operate unchanged on genuine evidence, while dedicating a specialized model to adversarially manipulated inputs.

V.1 MODEL OVERVIEW

The adversarial detection model is an attention-based convolutional neural network tailored to raw byte sequences. It consists of a byte embedding layer followed by a stack of three convolutional blocks that integrate spatial and channel attention mechanisms, a global pooling layer, and two fully connected layers for binary classification. The total capacity is approximately 1.31 million parameters, resulting in a compact model suitable for deployment as a preprocessing stage in forensic pipelines.

V.2 TRAINING CONFIGURATION

Training is conducted on the full 30.72 million-fragment benchmark, which includes 768,000 clean fragments and 29.952 million adversarial fragments spanning all 39 attack variants. The dataset is split into 70% training, 15% validation, and 15% test partitions using stratified sampling to preserve the clean-adversarial ratio in each split. The model is trained with the Adam optimizer, binary cross-entropy loss, and a batch size of 128, using class-balanced batches to avoid collapse to the majority (adversarial) class. Early stopping with a patience of five epochs is configured based on validation accuracy, although training in practice runs the full schedule due to stable validation performance. The dataset contains substantially more adversarial fragments than clean fragments because each clean fragment is transformed using all 39 attack variants. This enumeration of attack variants is necessary to ensure comprehensive coverage of the attack space and to evaluate robustness across a wide range of perturbation strategies. As a result, the raw dataset exhibits a 39:1 adversarial-to-clean imbalance. However, this imbalance does not affect the training dynamics because the model is trained using class-balanced mini-batches that sample equal numbers of clean and adversarial fragments. This approach allows the dataset to fully represent the diversity of attack variants while preventing the classifier from developing a bias toward the adversarial class during optimization.

V.3 DETECTION PERFORMANCE

On the 4.6 million-sample test set, the adversarial detection model achieves 95.46% specificity on clean fragments and 91.34% recall on adversarial fragments, with an F1-score of 0.9542 and AUC-ROC of 0.9819. The overall accuracy is 91.44%; however, because the test set reflects the natural benchmark composition (97.5% adversarial, 2.5% clean), this metric is dominated by recall performance. Specificity and recall are therefore reported as the primary evaluation metrics, as they independently capture detection quality on each class regardless of class distribution. Performance remains consistently strong across all attack categories. Padding-based attacks achieve the highest detection accuracies (93.47% for null-byte padding and 92.91% for random-byte padding), reflecting the strong boundary artefacts they introduce. Cross-file padding is the most challenging category yet still attains 86.24% accuracy. Random and header-targeted bit-flip attacks both exceed 90% detection accuracy, indicating that the learned byte embeddings and attention mechanisms capture even subtle changes in Hamming distance.

V.4 ROLE IN THE TWO-STAGE PIPELINE

Within the overall system, the adversarial detection model serves as a high-specificity front end that enables a detect-then-classify architecture without modifying the baseline ByteRCNN classifier. Clean fragments are classified by the original model at its full 71.1% accuracy, while adversarial fragments are redirected to the robust classifier that is explicitly trained to handle corrupted inputs. Misclassified adversarial fragments still receive baseline protection through ByteRCNN rather than catastrophic failure, and misclassified clean fragments experience only a minor accuracy penalty when processed by the robust classifier. This routing behavior is central to achieving strong robustness gains with a negligible impact on clean performance in the complete pipeline.

VI. ROBUST FILE FRAGMENT CLASSIFICATION DSAR-CNN MODEL

The vulnerability assessment in Section 4 established that ByteRCNN suffers a 55.3 pp accuracy collapse (71.1% → 15.8%) under adversarial attack. This catastrophic degradation raises a fundamental question: what architectural properties enable classification despite substantial byte-level corruption? The key insight motivating DSAR-CNN is that adversarial perturbations do not destroy all discriminative information. A 512-byte fragment subjected to 256-byte null-byte padding has its final 256 bytes replaced with zeros, while retaining 256 authentic bytes in the first half, and it often includes critical file-type markers (e.g., PDF headers, JPEG markers). Bit-flip attacks, despite corrupting individual bytes, typically affect only 1–20% of the fragment. The remaining authentic content retains sufficient structure to enable file-type identification if the classifier can selectively focus on un-corrupted regions and suppress noise from perturbed areas. Traditional CNNs fail under this scenario because they process all regions equally, allowing corrupted bytes to pollute global feature representations. DSAR-CNN addresses this through five synergistic innovations: multi-scale feature extraction to detect perturbations at matched granularities, learnable position embeddings to recognize attack-specific positional patterns, hierarchical attention mechanisms to separate file-type signals from adversarial noise, split pooling to resist attention-induced feature dilution, and regularized classification to prevent overfitting to specific attack patterns.

VI.1 DSAR-CNN ARCHITECTURE OVERVIEW

DSAR-CNN comprises six layers designed to progressively extract, refine, and classify features from corrupted byte sequences. The complete model contains 479,344 parameters and achieves 68.4% accuracy on adversarial fragments when evaluated on the final test set (4.6 million samples). (Note: Ablation study results on the validation set show 70.1% to isolate individual component contributions; see Table 4). Figure 2 shows the detailed architecture of the DSAR-CNN classification model.

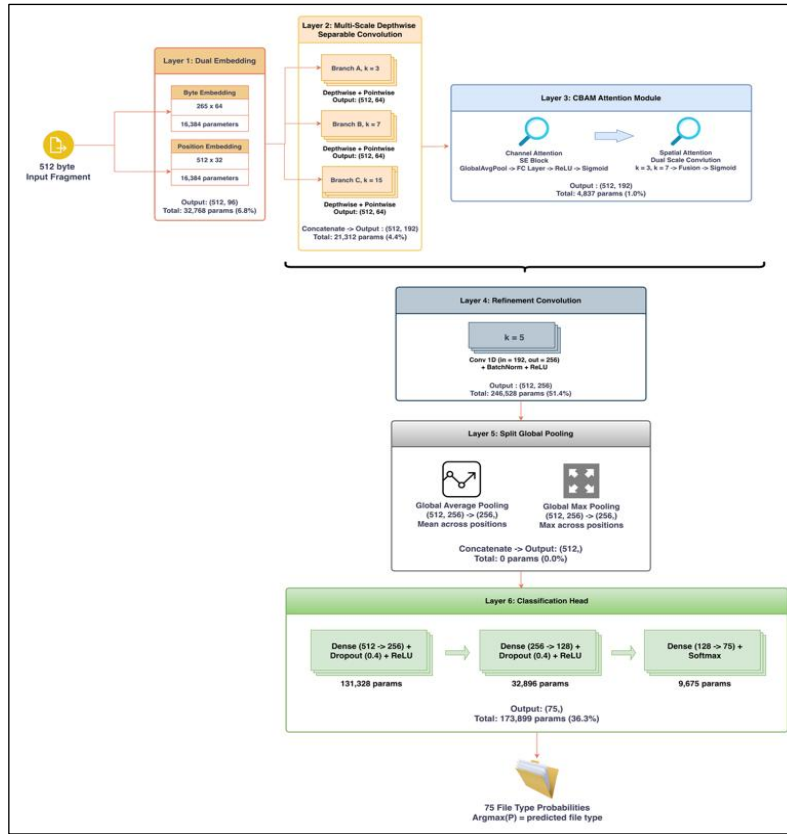


Figure 2: Proposed Dual-Scale Attention-based Robust CNN (DSAR-CNN) architecture for adversarial file fragment classification. Source: Authors, (2026).

VI.1.1 Layer 1: Dual Embedding

Layer 1 transforms raw byte integers (0–255) into rich, position-aware feature vectors that capture both semantic byte meaning and contextual positional information. This dual representation is crucial because adversarial attacks create position-dependent corruption patterns. Byte Embedding. Each byte value is mapped to a learnable 64-dimensional embedding:

$$X_{\text{byte}}[i] = E[b_i, :] \text{ for } i \in [0, 511] \quad (9)$$

where $E \in \mathbb{R} \{256 \times 64\}$ is the learnable embedding matrix and $b_i \in \{0, 1, \dots, 255\}$ is the byte value. This creates semantic representation: byte value 0 learns associations with padding, specific ASCII characters learn their characteristic file context, and so forth. Parameters: $256 \times 64 = 16,384$. Learnable Position Embedding. Each position is mapped to a learnable 32-dimensional embedding:

$$X_{\text{pos}}[i] = P[i, :] \text{ for } i \in [0, 511] \quad (10)$$

where $P \in \mathbb{R} \{512 \times 32\}$ is the learnable position matrix. These embeddings are trained end-to-end on adversarial data. During training, the model learns that tail positions (448–511) frequently contain padding artifacts while earlier positions are more likely authentic. Parameters: $512 \times 32 = 16,384$. Concatenation. The two embeddings are concatenated along the feature dimension:

$$X_{\text{embed}}[i] = [X_{\text{byte}}[i] \parallel X_{\text{pos}}[i]] \in \mathbb{R}^{96} \quad (11)$$

Output shape: (512, 96). Each position now has a 96-dimensional vector capturing both what byte it is and where it is located.

VI.1.2 Layer 2: Multi-Scale Depth-wise Separable Convolution

Purpose. Different adversarial attacks create perturbation patterns at fundamentally different spatial scales: bit-flip attacks corrupt individual bytes (1–20 bytes), while padding attacks create large contiguous blocks (16–256 bytes). A single convolution kernel cannot optimally detect both. Layer 2 uses three parallel branches with kernel sizes matched to attack granularities. The kernel sizes (3, 7, and 15) were selected to provide multi-scale receptive fields capable of capturing both local byte-level variations and broader contextual patterns within the fragment. Smaller kernels capture fine-grained byte transitions, while larger kernels capture longer-range contextual patterns such as padding boundaries or abrupt distribution changes. This multi-scale design enables the model to analyze perturbations occurring at different spatial extents. Much larger kernels (e.g., 85, 125) would unnecessarily dilute these localized signals, increase parameter count, and risk overfitting without providing additional discriminative power beyond what is already achieved through hierarchical stacking of smaller convolutions. Three Branches. The layer processes the 512-position sequence with three 1D convolution branches simultaneously:

1. Branch A (k=3): Fine-grained analysis for scattered bit-flip perturbations
2. Branch B (k=7): Medium-scale analysis for local pattern disruptions
3. Branch C (k=15): Coarse-grained analysis for padding attack boundaries

Each branch produces 64 output channels, which are concatenated to form 192 total channels. Depthwise Separable Convolution. Instead of standard convolution, each branch uses depthwise separable convolution, which factorizes the operation into two steps: (1) depthwise convolution applies a separate kernel to each of the 96 input channels independently, and (2) pointwise (1×1) convolution mixes information across channels. Parameter count also includes biases on both depthwise and pointwise convolutions. Parameter Calculation. For a fair comparison including biases, a standard convolution across the three kernels would use:

$$(k_1 + k_2 + k_3) \times C_{\text{in}} \times C_{\text{out}} + 3 \times C_{\text{out}} = 25 \times 96 \times 64 + 3 \times 64 = 153,792 \text{ params} \quad (12)$$

Depthwise separable achieves:

$$\sum_{j=1}^3 (k_j C_{\text{in}} + C_{\text{in}} + C_{\text{in}} C_{\text{out}} + C_{\text{out}}) = 21,312 \text{ params} \quad (13)$$

VI.1.3 Layer 3: CBAM Attention Module

Purpose. After extracting multi-scale features, Layer 3 learns which features are reliable (file-type discriminative) and which are noisy (adversarial artifacts). It does this through two complementary attention mechanisms: channel attention identifies important feature channels and spatial attention identifies trustworthy byte positions. Channel Attention (Squeeze-and-Excitation). This mechanism first computes the average activation of each of the 192 feature channels across all positions, then learns channel importance weights $s \in (0,1)^{\{192\}}$:

$$z = \text{GlobalAvgPool}(F) \in \mathbb{R}^{192} \quad (14)$$

$$s = \sigma(W_2 \cdot \text{ReLU}(W_1 \cdot z)) \in (0,1)^{192} \quad (15)$$

where σ is sigmoid, $W_1 \in \mathbb{R}^{\{12 \times 192\}}$ maps to a 12-dimensional bottleneck (reduction ratio $r=16$), and $W_2 \in \mathbb{R}^{\{192 \times 12\}}$ reconstructs channel weights. These weights amplify channels encoding file-type patterns (weight ≈ 0.8 – 0.95) and suppress channels encoding adversarial noise (weight ≈ 0.05 – 0.2). Parameters: $(192 \times 12) + 12 + (12 \times 192) + 192 = 4,608$ weights + 204 biases = 4,812. Spatial Attention (Dual-Scale). Operating on the channel-refined features, spatial attention asks which positions are trustworthy. It uses two scales ($k=3$ and $k=7$) to detect corruption at different granularities:

$$D = [\text{AvgPool}_{\text{channels}}(F); \text{MaxPool}_{\text{channels}}(F)] \quad (16)$$

$$\alpha = \sigma(W_f \cdot [\text{Conv}(D, k=3); \text{Conv}(D, k=7)]) \in (0,1)^{512} \quad (17)$$

The 2-channel descriptor D (formed by channel-wise average and max pooling) is processed by two parallel 1D convolutions ($k=3$ and $k=7$, each $2 \rightarrow 1$ channels with bias), whose outputs are concatenated and fused by a 1×1 convolution ($2 \rightarrow 1$ channels with bias). Position-wise weights α suppress corrupted regions ($\alpha \approx 0.05$ – 0.15) and preserve authentic regions ($\alpha \approx 0.9$ – 0.95). Parameters: $(2 \times 3 + 1) + (2 \times 7 + 1) + (2 \times 1 + 1) = 7 + 15 + 3 = 25$. Synergistic Effect. An interesting finding is that channel attention + spatial attention produces +11.2 pp improvement in adversarial accuracy. This synergy arises from hierarchical structure: channel attention first filters out noisy channels, producing a cleaner signal that spatial attention then operates on. Operating on this pre-filtered signal, spatial attention makes more reliable position-level decisions.

VI.1.4 Layer 4: Refinement Convolution

Purpose. By Layer 4, features have been extracted from three parallel perspectives ($k=3$, $k=7$, $k=15$ branches) and selectively refined by CBAM attention mechanisms. Layer 4's role is to integrate this diverse multi-scale information into a unified, higher-dimensional representation. It acts as a synthesis layer that combines the complementary perspectives before passing to the classification head. At the input to Layer 4, the model has 512 positions, each with 192 feature dimensions. These 192 dimensions are organized in three groups of 64 channels, each representing patterns detected at different scales:

- Channels 0–63: Fine-grained patterns from $k=3$ analysis (subtle byte-level corruption signatures)
- Channels 64–127: Medium-scale patterns from $k=7$ analysis (local disruption boundaries)
- Channels 128–191: Coarse-grained patterns from $k=15$ analysis (large padding block transitions)

Additionally, spatial attention has already suppressed positions suspected of containing adversarial perturbations. Authentic regions (typically positions 0–383 in padding attacks) have attention weights $\alpha \approx 0.9$ –0.95, while corrupted regions (typically positions 384–511) have been suppressed to $\alpha \approx 0.05$ –0.15. The network now has a partially masked but information-rich feature representation. Operation. A 1D convolution with $k=5$ processes the attended features

$$F_{\text{refined}} = \text{ReLU} \left(\text{BatchNorm}(\text{Conv1D}(F_{\text{attended}}, k = 5, 256 \text{ filters})) \right) \quad (18)$$

Layer 4 applies a 1D convolution with $k=5$ that allows neighboring positions (± 2 away) to collectively influence each output position. This local context aggregation captures file-type patterns, which are often locally coherent; authentic regions maintain consistent byte patterns while corruption boundaries create sharp discontinuities. The convolution simultaneously expands channels from 192 to 256, learning 256 different perspectives including combinations like "fine-grained and medium-scale corruptions co-occurring" or "authentic header followed by noise." Batch normalization then normalizes activations to ensure stable training on the imbalanced dataset (97.5% adversarial, 2.5% clean), while ReLU introduces non-linearity to preserve the network's expressive power.

VI.1.5 Layer 5: Split Global Pooling

Purpose. Layers 1–4 maintain position-level information across 512 positions. Layer 5 compresses this into a single global representation but uses split pooling to counteract a critical problem: dilution from attention-based suppression. The Dilution Problem. After CBAM attention, many positions in adversarially perturbed fragments have been suppressed to near-zero activations. For a fragment under 256-byte padding attack, spatial attention suppresses positions 256–511 ($\alpha \approx 0.05$ –0.15), while preserving positions 0–255 ($\alpha \approx 0.9$ –0.95). When Global Average Pooling computes the mean across all 512 positions:

$$f_{\text{gap}} = \frac{1}{512} \sum_{i=1}^{512} F_{\text{refined}}[i] \quad (19)$$

The result averages 256 strong authentic activations with 256 near-zero suppressed activations, producing approximately 50% of the signal strength compared to a fully clean fragment. This dilution - averaging real signals with suppressed noise - weakens the discriminative information available to the classifier. A classifier receiving a 50%-diluted representation of a PDF fragment might have difficulty distinguishing it from a JPEG, even if the authentic 50% contains clear PDF signatures. Split Pooling Solution. Rather than accepting this dilution, Layer 5 employs two complementary pooling operations: Global Average Pooling (GAP) is defined as:

$$f_{\text{gap}} = \frac{1}{512} \sum_{i=1}^{512} F_{\text{refined}}[i] \in \mathbb{R}^{256} \quad (20)$$

GAP is sensitive to position suppression, making it diluted on corrupted fragments but representative on clean fragments. Global Max Pooling (GMP) is defined as:

$$f_{\text{gmp}} = \max_i F_{\text{refined}}[i] \in \mathbb{R}^{256} \quad (21)$$

GMP finds the maximum activation across all 512 positions. Crucially, GMP is un-affected by suppression - even if 256 positions are suppressed to near-zero, the maximum activation remains strong from the 256 unsuppressed positions. GMP preserves the strongest signals regardless of position distribution. Concatenation and Interpretation. The classifier receives both:

$$f_{\text{pooled}} = [f_{\text{gap}} \parallel f_{\text{gmp}}] \in \mathbb{R}^{512} \quad (22)$$

The classification head receives both the Global Average Pooling (GAP) and Global Max Pooling (GMP) representations, allowing it to leverage complementary information about the fragment. GAP summarizes the overall activation distribution across positions and is therefore sensitive to the suppression of corrupted regions introduced by spatial attention. In contrast, GMP preserves the strongest activation within the fragment and remains robust even when many positions have been suppressed. When large portions of the fragment are corrupted, GAP values tend to decrease due to averaging over suppressed positions, while GMP continues to reflect the strongest remaining signals. By concatenating these two representations, the classifier receives both a global summary of the fragment and the strongest local activation patterns, providing richer information for classification under corrupted conditions. Why This Matters. Using only Global Average Pooling would produce diluted feature representations when large regions of the fragment are suppressed by attention, potentially weakening discriminative signals.

Conversely, Global Max Pooling alone would ignore the broader distribution of activations across the fragment. Combining GAP and GMP allows the model to retain both strong local activations and global contextual information, improving robustness when fragments contain partially corrupted regions. The ablation study confirms the effectiveness of this design, showing that removing split pooling reduces adversarial accuracy by 6.3 percentage points.

VI.1.6 Layer 6: Classification Head

Purpose. The final three fully connected layers map the 512-dimensional pooled features to 75-class file-type probabilities. By this stage, features have been carefully extracted and refined; the classification task is relatively straightforward mapping of rich representations to file-type labels. **Architecture.** The classification head consists of three dense layers: Dense1 (512→256 with ReLU + dropout), Dense2 (256→128 with ReLU + dropout), and Output (128→75 with softmax). This gradual dimensionality reduction from 512 to 75 prevents sudden compression that could discard information, allowing the model to progressively focus features on file-type discrimination. **Dropout** with probability 0.4 is applied after each ReLU activation, where 0.4 was selected based on preliminary hyperparameter tuning across rates of 0.3, 0.4, and 0.5. **Label Smoothing.** During training, label smoothing with smoothing factor 0.1 is applied. Instead of training on hard labels (e.g., [0,0,...,1,...,0] for PDF), the model trains on smoothed targets where the correct class receives probability 0.9 and other 74 classes share 0.1 probability equally. This prevents overconfidence on corrupted patterns, particularly important given the 97.5% adversarial sample composition in training data. Without smoothing, the model might learn to confidently predict incorrect file types when seeing corrupted byte patterns, reducing calibration. The Figure 3 shows the parameter summary for the DSAR-CNN model.

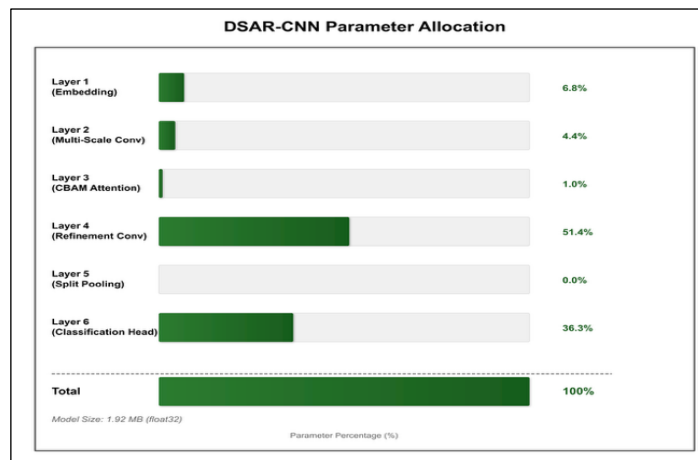


Figure 3: Parameter distribution in proposed DSAR-CNN model.
Source: Authors, (2026).

VI.2 ABLATION STUDY: COMPONENT CONTRIBUTIONS

To quantify the contribution of each architectural component, an ablation study was conducted in which individual components were removed from the full DSAR-CNN architecture and the resulting models were retrained from scratch. For each variant (e.g., without Position Embedding), the specified component was removed while all other architectural elements remained unchanged. Early stopping based on validation performance typically halted training before the maximum epoch limit, reducing overall training time while preserving fair comparisons across variants. When a component was removed, it was replaced with the simplest functional alternative required to maintain architectural compatibility (e.g., removing split pooling results in standard global average pooling). Model performance was then evaluated on the validation set using adversarial fragments. The contribution of each component is estimated as the difference between the full model accuracy (70.1%) and the accuracy of the corresponding ablated variant. Since all other experimental conditions are held constant, the observed performance differences primarily reflect the contribution of the removed component. Table 4 summarizes the results of this analysis.

Table 4: Ablation Study of DSAR-CNN Components: Adversarial Accuracy and Marginal Contribution on Validation Set.

Configuration	Adversarial Accuracy	Accuracy Drop
Full DSAR-CNN	70.10%	-
Without Position Embedding	65.30%	-4.8 pp
Without Multi-Scale Conv	62.80%	-7.3 pp
Without Channel Attention	66.50%	-3.6 pp
Without Spatial Attention	67.20%	-2.9 pp
Without CBAM (channel + spatial attention)	58.90%	-11.2 pp
Without Split Pooling	63.80%	-6.3 pp

Source: Authors, (2026).

This table reports accuracy on the validation set during model development to estimate the relative impact of each architectural component. The full DSAR-CNN achieves 70.1% accuracy on the validation set used for ablation analysis. Final evaluation on the independent test set (4.6M fragments) yields 68.4% accuracy, reflecting normal generalization differences between development and test datasets. The relative contribution percentages remain valid because they measure deltas between models trained and evaluated in the same setting (validation set).

Results show that different components contribute differentially. The combined CBAM module produces the largest performance drop when removed (-11.2 pp), followed by multi-scale convolution (-7.3 pp) and split pooling (-6.3 pp). Position embedding ($+4.8$ pp), channel attention ($+3.6$ pp), and spatial attention ($+2.9$ pp) provide smaller but still meaningful contributions. CBAM Synergy. The combined CBAM module provides a larger improvement ($+11.2$ pp) than the sum of the individual channel ($+3.6$ pp) and spatial ($+2.9$ pp) attention contributions. This indicates that the two attention mechanisms interact synergistically, where channel attention first suppresses noisy feature channels and spatial attention subsequently identifies reliable byte positions within the refined feature representation. Without this hierarchy, spatial attention operating on raw, unfiltered channels may find it harder to distinguish corruption from genuine patterns, which can limit its effectiveness. This hierarchical synergy is consistent with observations reported in prior attention mechanism studies and demonstrates that component interactions can produce amplified benefits beyond independent contributions.

VI.3 TRAINING CONFIGURATION

DSAR-CNN is trained on the 70% training allocation: 20,966,400 adversarial samples and 537,600 clean samples, totaling 21,504,000 fragments. The 2.50% clean sample inclusion, though minimal, helps DSAR-CNN learn robust features when detector false positives (FPR=4.54%) route clean fragments incorrectly to this branch. Hyperparameters: Adam optimizer (learning rate 0.001, reduced on plateau by factor 0.5), sparse categorical cross-entropy loss, batch size 256, early stopping (5-epoch patience), dropout 0.4, label smoothing 0.1, batch normalization, high momentum (0.99) to obtain stable running statistics given the large, highly skewed training distribution. Training converged within 35–40 epochs (~ 25 – 30 hours on modern GPU).

VI.4 PERFORMANCE SUMMARY

Standalone DSAR-CNN Performance (Test Set Results): DSAR-CNN achieves a 68.4% accuracy on adversarially perturbed fragments. And DSAR-CNN achieves 67.0% accuracy on clean fragments (compared to 71.1% for ByteRCNN), reflecting the trade-off inherent in training primarily on adversarial data.

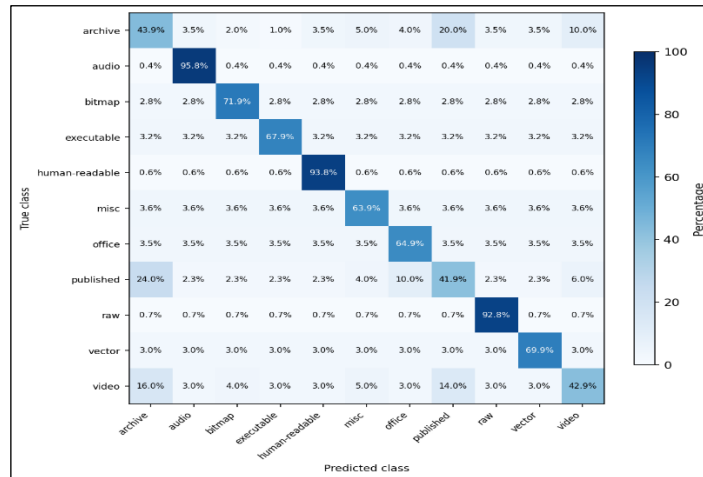


Figure 4: Confusion matrix of DSAR-CNN evaluated on the full test set. Given the large number of classes (75), individual file types were aggregated into their corresponding superclasses.

Source: Authors, (2026).

The confusion matrix in Figure 4 is computed over the test set of 4.6M fragments, which is dominated by adversarial samples. The average class-wise accuracy ($\sim 68\%$) is consistent with the reported adversarial accuracy of 68.4%. The performance follows typical FFC difficulty trends, with structured formats such as audio, text, and raw data showing high robustness, while high-entropy formats such as archives, published content, and video remain more challenging. The observed confusion patterns between these entropy-rich formats further reflect realistic behavior under adversarial conditions. The large robustness gain over ByteRCNN arises because DSAR-CNN is explicitly optimized for the same padding and bit-flip transformations used at test time, whereas ByteRCNN is trained only on clean fragments and therefore fails under severe distribution shift. By training on a large, supervised corpus of perturbed fragments and incorporating architectural bias toward perturbation-resilient features (multi-scale depthwise-separable convolutions, attention, and split pooling), DSAR-CNN learns to focus on the remaining informative bytes and down-weight corrupted regions. In this matched train–test setting, an improvement in adversarial accuracy over a clean-only baseline is expected and does not imply any form of data leakage.

VI.4.1 Attack Category-wise Performance

DSAR-CNN achieves substantial improvements across all attack categories. On padding attacks (15 variants), the model attains an average accuracy of 67.1%, compared to ByteRCNN's 11.2% baseline. Null-byte and random-byte padding are handled more effectively (68.4% and 67.8%) due to the clear boundary artifacts they introduce, while cross-file padding is more challenging (65.2%) because it contains authentic but contextually incorrect byte sequences from other file types. For bit-flip attacks, random bit-flip variants achieve an average accuracy of 70.0% versus ByteRCNN's 20.2%. Header-targeted bit-flip attacks reach 68.3% compared to 17.1%. These results indicate that DSAR-CNN provides strong robustness across both distributed and localized corruption patterns.

VI.5 ROLE IN TWO-STAGE PIPELINE

Within the complete defense system, DSAR-CNN serves as the specialized stage-2 classifier for adversarially perturbed fragments. When the adversarial detector flags a fragment as adversarial (91.34% recall), it is routed to DSAR-CNN rather than the baseline classifier. This routing mechanism enables preservation of baseline accuracy on clean data while achieving substantial robustness improvements. Fragments misclassified by the detector (false negatives) still receive some protection through ByteRCNN's partial robustness, while detector false positives (clean fragments routed to DSAR-CNN) experience modest accuracy reduction due to the 2.5% clean training data inclusion.

VII. EXPERIMENTAL EVALUATION AND PROPOSED FRAMEWORK PERFORMANCE

The complete defense system integrates three models into a sequential two-stage pipeline. A 512-byte fragment first passes through the adversarial detection model, which classifies it as clean or adversarial with 91.44% accuracy, 91.34% recall, and 95.46% specificity. Fragments predicted as clean are routed to ByteRCNN (preserving 71.1% baseline accuracy), while fragments flagged as adversarial are routed to DSAR-CNN (achieving 68.4% standalone adversarial accuracy). This routing mechanism enables the system to maintain high accuracy on genuine evidence while substantially improving robustness on perturbed inputs.

VII.1 END-TO-END PIPELINE PERFORMANCE

VII.1.1 Clean Data Performance

On 115,200 clean test fragments, the pipeline achieves 70.91% accuracy, a 0.19 pp penalty compared to the 71.1% undefended baseline. This metric is derived from the weighted average of clean fragments correctly routed to ByteRCNN (True Negatives) and those misrouted to DSAR-CNN (False Positives):

$$Acc_{clean}^{pipeline} = (Acc_{clean}^{Byte} \times \text{Specificity}) + (Acc_{clean}^{DSAR} \times \text{FPR}) \quad (23)$$

Substituting the experimental values:

$$70.91\% \approx (71.1\% \times 0.9546) + (67.0\% \times 0.0454) \quad (24)$$

The minimal penalty demonstrates that detection-based defense enables preservation of baseline accuracy on genuine evidence while adding robustness to adversarial inputs.

VII.1.2 Adversarial Data Performance

On 4,492,800 adversarial test fragments across all 39 attack variants, the pipeline achieves 63.84% accuracy, compared to 15.8% for the undefended baseline. The pipeline accuracy is calculated as the weighted sum of adversarial fragments correctly routed to DSAR-CNN (True Positives) and those misclassified as clean (False Negatives):

$$Acc_{adv}^{pipeline} = (Acc_{adv}^{DSAR} \times \text{Recall}) + (Acc_{adv}^{Byte} \times \text{FNR}) \quad (25)$$

Substituting the detector performance:

$$63.84\% \approx (68.4\% \times 0.9134) + (15.8\% \times 0.0866) \quad (26)$$

This calculation confirms that the system's robustness is primarily driven by the robust classifier (DSAR-CNN), with graceful degradation provided by the baseline classifier for the 8.66% of undetected attacks.

VII.1.3 Mixed-Threat Scenario (90% Clean / 10% Adversarial)

In operational cybersecurity systems, malicious activity typically constitutes a small minority of total observations, a trend documented in large-scale intrusion detection studies such as CICIDS-2017 and related imbalance analyses by Buczak and Guven and Zhao et al [35-37]. For instance, the CICIDS-2017 benchmark - one of the most widely used intrusion detection datasets - contains predominantly benign traffic with attack flows forming only a minority of total connections, and real-world deployment reports indicate even more skewed benign-to-attack ratios in production networks [35]. Although no published adversarial prevalence rates exist specifically for forensic fragment recovery scenarios, the 90/10 clean-to-adversarial ratio adopted here therefore reflects a realistic attack-sparse deployment assumption that is consistent with empirical observations from these adjacent security domains [36] [37]. The pipeline achieves 70.20% accuracy on a typical mixed dataset (90% clean, 10% adversarial). This metric combines the pipeline's clean and adversarial performance calculated above:

$$Acc_{mixed} = (Acc_{clean}^{pipeline} \times P_{clean}) + (Acc_{adv}^{pipeline} \times P_{adv}) \quad (27)$$

$$70.20\% = (70.91\% \times 0.90) + (63.84\% \times 0.10) \quad (28)$$

This represents a 4.63 pp improvement over the undefended baseline mixed accuracy, which is calculated similarly:

$$Acc_{mixed}^{baseline} = (71.1\% \times 0.90) + (15.8\% \times 0.10) = 65.57\% \quad (29)$$

Operationally, this improvement translates to approximately 46 additional correctly classified fragments per 1,000 processed - a meaningful enhancement for large-scale forensic investigations.

VII.2 COMPARISON WITH ADVERSARIAL TRAINING AND PIPELINE PERFORMANCE METRICS SUMMARY

To establish a direct comparison with the dominant defense paradigm, standard adversarial training was implemented on the baseline ByteRCNN architecture using a balanced 50/50 clean-adversarial training mix, following the standard adversarial training protocol [24]. The training dataset comprised 1,075,200 samples: 537,600 clean fragments and 537,600 adversarial fragments sampled uniformly across all 39 attack variants (approximately 13,784 samples per variant). Adversarial samples were generated exclusively from the training partition, and all training configurations (architecture, optimizer, and hyperparameters) were kept identical to the baseline to ensure a fair architectural comparison. The balanced ratio inherently constrains training volume to 1,075,200 samples, compared to the 21,504,000-fragment set used by the defense pipeline at its natural 39:1 adversarial-to-clean ratio. This asymmetry reflects a fundamental distinction between the two paradigms: adversarial training requires balanced class exposure to avoid majority-class collapse, whereas detection and robust classification models are designed to operate on realistic imbalanced distributions. Scaling adversarial training to match pipeline volume at 50/50 balance would require 20× oversampling of the clean fragment pool, introducing artificial duplication rather than genuine training signal. The adversarially trained model was evaluated on the identical test set of 4,608,000 fragments used throughout this study. Table 6 presents the comparative results.

Table 5: Comparative Performance Metrics of the Proposed Pipeline Against Baseline and Adversarial Training Approaches.

Metric	Clean	Adversarial	Mixed (90/10)
Undefended baseline	71.10%	15.80%	65.57%
Adversarial Training (50/50)	66.60%	51.80%	65.12%
Defense pipeline (Proposed)	70.91%	63.84%	70.20%
Improvement (Proposed vs. Adv. Training)	+4.31 pp	+12.04 pp	+5.08 pp

Source: Authors, (2026).

Adversarial training does improve robustness, increasing adversarial accuracy from 15.8% to 51.8% (a gain of 36.0 percentage points). However, this improvement comes at the cost of a 4.5 percentage point reduction in clean accuracy (from 71.1% to 66.6%), confirming the accuracy-robustness trade-off widely reported in adversarial learning literature. A critical observation emerges from the mixed-scenario analysis. In realistic forensic environments where 90% of fragments are benign and 10% are adversarially perturbed, adversarial training achieves a weighted accuracy of 65.12%, which is slightly lower than the undefended baseline (65.57%). This occurs because the reduction in clean accuracy outweighs the robustness gains when adversarial samples constitute a minority of the data. The proposed detection-based pipeline achieves superior performance across all evaluation settings. Compared to adversarial training, the pipeline improves clean accuracy by 4.31 percentage points (70.91% vs. 66.6%), adversarial accuracy by 12.04 percentage points (63.84% vs. 51.8%), and mixed-scenario accuracy by 5.08 percentage points (70.20% vs. 65.12%). These results demonstrate that the detect-then-classify routing strategy effectively mitigates the accuracy-robustness trade-off by preserving baseline performance on clean inputs while improving robustness against adversarial fragments. All models were trained once due to computational constraints. However, the large test set size (4.6M fragments) yields narrow confidence intervals: the 95% CI for the pipeline's mixed accuracy of 70.20% is [70.16%, 70.24%] and for adversarial training's 65.12% is [65.08%, 65.16%], confirming the 5.08 pp difference is statistically significant ($p < 0.001$, two-proportion z-test). The training volume differs between adversarial training and the proposed pipeline, reflecting differences in how the two approaches are typically implemented. In adversarial training, balanced clean-adversarial datasets are commonly used to maintain stable optimization and avoid oversampling artifacts when scaling to large attack sets. In contrast, the proposed pipeline leverages the full adversarial dataset through a detection-routing mechanism. Consequently, the results should be interpreted as a comparison between the proposed framework and adversarial training under its standard configuration rather than a strictly matched-volume experimental setting. To ensure that the reported results are not contingent on this specific operating point, Table 6 provides a sensitivity analysis across a range of adversarial prevalence rates (p). The formula for each cell is:

$$Acc_{mixed} = (Acc_{clean} \times (1 - p)) + (Acc_{adv} \times p) \quad (30)$$

Table 6: Sensitivity of Pipeline and Adversarially Trained Baseline Accuracies to Adversarial Prevalence.

Adversarial Prevalence	Pipeline Accuracy	AT Baseline	Pipeline Lead over AT
0% (all clean)	70.91%	66.60%	+4.31 pp
5%	70.56%	65.86%	+4.70 pp
10% (primary)	70.20%	65.12%	+5.08 pp
20%	69.50%	63.64%	+5.86 pp
30%	68.79%	62.16%	+6.63 pp
50% (balanced)	67.38%	59.20%	+8.18 pp

Source: Authors, (2026).

The proposed pipeline outperforms the adversarial training baseline across all tested prevalence rates from 0% to 50%. Notably, the pipeline's advantage over adversarial training increases with adversarial prevalence (from +4.31 pp at 0% to +8.18 pp at 50%), demonstrating that the detection-routing mechanism provides progressively greater benefit as adversarial activity increases."

VII.3 ROBUSTNESS BY ATTACK CATEGORY

Performance across individual attack categories demonstrates that the proposed pipeline provides consistent robustness improvements over both the undefended baseline and the adversarially trained model. The undefended baseline is included to quantify the severity of adversarial degradation and to show the extent of robustness recovery. However, the primary comparison for defense effectiveness is against adversarial training, which represents the standard defense approach. Figure 5 visualizes these per-category results.

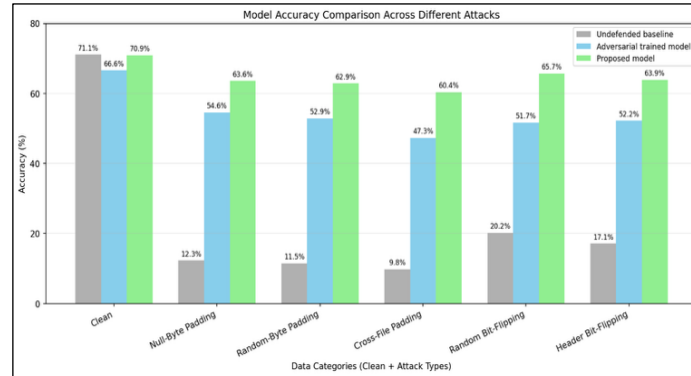


Figure 5: Model accuracy comparison across attack categories for the undefended baseline, adversarial training and proposed pipeline. Source: Authors, (2026).

On clean data, the proposed pipeline achieves an accuracy of 70.9%, which is comparable to the baseline performance of 71.1% and substantially higher than the adversarially trained model (66.6%). This result confirms that the detection-based routing strategy preserves clean-data performance and avoids the accuracy degradation typically associated with adversarial training. Under padding-based attacks, which introduce large structural perturbations and are the most damaging to the baseline model, the proposed approach maintains strong and stable performance. For null-byte padding, accuracy improves from 12.3% (baseline) and 54.6% (adversarial training) to 63.6%. Random-byte padding shows a similar trend, increasing from 11.5% and 52.9% to 62.9%, while cross-file padding improves from 9.8% and 47.3% to 60.4%. These results represent improvements of approximately 8-13 percentage points over adversarial training. Although cross-file padding remains the most challenging category due to the insertion of realistic but contextually misleading byte patterns, the proposed pipeline maintains substantially higher robustness than both comparison models. For bit-flip attacks, which introduce localized corruption rather than large-scale structural changes, the proposed approach again achieves the highest performance. Random bit-flip accuracy increases from 20.2% (baseline) and 51.7% (adversarial training) to 65.7%, while header-targeted bit-flips improve from 17.1% and 52.2% to 63.9%. The gains over adversarial training range from approximately 11 to 14 percentage points, indicating that the detect-then-classify strategy effectively isolates perturbed inputs and enables more accurate classification through the specialized DSAR-CNN model. Two consistent trends emerge from this analysis. First, padding-based perturbations cause more severe degradation in the undefended baseline than bit-flip attacks, reflecting their larger impact on the underlying fragment structure and statistical distribution. Second, the proposed pipeline delivers stable and balanced performance across all attack categories, outperforming adversarial training while maintaining near-baseline accuracy on clean data. This behavior demonstrates that the approach effectively improves robustness without incurring the accuracy-robustness trade-off commonly observed in conventional adversarial training approaches.

VIII. DISCUSSION AND FUTURE WORK

VIII.1 SIGNIFICANCE AND ARCHITECTURAL ADVANTAGES

This work addresses two important limitations in current file fragment classification research. First, it provides a systematic vulnerability analysis of state-of-the-art FFC models under byte-level perturbations. Second, it explores a detection-based defense strategy designed to improve robustness while preserving clean-data accuracy: 70.91% clean accuracy (−0.19 pp penalty) with 63.84% adversarial accuracy, contrasting sharply with the 4.5% clean accuracy penalty empirically observed in the adversarial training baseline (Section 7.2). The architecture is specifically tailored to byte-level FFC. Multi-scale depthwise separable convolution ($k=3, 7, 15$) matches attack granularities: $k=3$ for scattered bit-flips, $k=7$ for local disruptions, $k=15$ for padding boundaries. This scale-matching is unnecessary in image defenses where perturbations are uniform. Split pooling addresses byte-level dilution: when spatial attention suppresses corrupted positions, GMP preserves strongest signals while GAP measures dilution, enabling the classifier to infer corruption implicitly. Learnable position embeddings exploit positional attack structure (tail-biased padding) more efficiently than fixed encodings.

VIII.2 LIMITATIONS AND FUTURE WORK

Evaluation Limitations. The current evaluation focuses on black-box attacks. White-box adaptive attacks where adversaries have full model access are not tested. In theory, attackers could exploit the learned position embeddings by shifting padding to the fragment head instead of the tail or target the attention mechanism directly. Certified defense approaches like randomized smoothing or IBP could address this but implementing them across 30 million samples is computationally expensive and remains as future work. Multi-seed experiments would better characterize model stability, and latency measurements are theoretical rather than measured across different hardware platforms. A key limitation of this evaluation is that each attack variant is applied independently to clean fragments, and do not study composite adversaries that combine padding and bit-flip perturbations within the same fragment, which may represent a stronger and more realistic threat model in operational settings.

Architectural Limitations. The position embeddings are specifically learned for tail-biased padding attacks. If adversaries instead corrupt the fragment head, the positional priors might degrade performance. The architecture was only tested on 512-byte fragments - generalization to 256-byte or 1024-byte sizes remains unclear. Cross-dataset evaluation on alternative benchmarks like DFRWS would be valuable. Finally, the 39 attack variants tested (padding and bit-flips) do not cover every possible byte-level perturbation strategy, so robustness to novel attacks is uncertain. **Future Directions.** Priority work includes testing head padding and random-position attacks to validate that position embeddings learn genuine file structure rather than memorizing attack positions; extending the detector to classify attack types for forensic attribution; developing certified robustness guarantees; evaluating on alternative datasets and fragment sizes; and integrating with commercial forensic tools for real-world validation. Modeling white-box adaptive attacks would also be important to understand worst-case robustness.

IX. CONCLUSIONS

File fragment classification remains essential to digital forensics, yet the vulnerability assessment uncovered that the state-of-the-art model accuracy collapses from 71.1% to 15.8% when facing byte-level adversarial perturbations. Testing across 39 attack variants (padding and bit-flip strategies) and five distinct attack categories reveals this vulnerability is systematic and universal, not confined to edge cases. The threat is real: simple, format-agnostic byte manipulations suffice to compromise classification integrity, rendering current forensic systems unreliable in adversarial contexts. The proposed two-stage detection-then-classify pipeline substantially mitigates this threat (70.91% clean, 63.84% adversarial, 70.20% in realistic mixed scenarios with 90% clean and 10% adversarial fragments). The improvement is consistent across all attack categories: 9 pp for null padding, 10 pp for random byte padding, 13.1 pp for cross-file padding, 14 pp for random bit-flip, and 11.7 pp for header-targeted bit-flip as compared to the adversarially trained model values. This recovery comes with minimal clean accuracy penalty (0.19 pp) too. The architectural innovations underlying this defense - multi-scale depthwise separable convolution, hierarchical CBAM attention, split pooling, and learnable position embeddings - are tailored to byte-sequence processing and extend beyond file fragment classification to malware detection, network traffic analysis, and other security domains relying on byte-level feature extraction. The detection-based routing strategy sidesteps the accuracy-robustness tradeoff plaguing adversarial training by preserving baseline performance on genuine evidence while routing adversarial inputs to specialized classifiers. These results demonstrate that detection-based routing strategies can substantially improve robustness against byte-level perturbations while preserving clean-data performance. The proposed framework therefore provides a promising direction for strengthening the resilience of file fragment classification systems against adversarial manipulation.

X. AUTHOR'S CONTRIBUTION

Conceptualization: Teena Mary and Sreeja C S.

Methodology: Teena Mary.

Investigation: Teena Mary.

Discussion of results: Teena Mary and Sreeja C S.

Writing – Original Draft: Teena Mary.

Writing – Review and Editing: Teena Mary and Sreeja C S.

Resources: Teena Mary and Sreeja C S.

Supervision: Sreeja C S.

Approval of the final text: Teena Mary and Sreeja C S.

XI. REFERENCES

- [1] M. Guzhov and C. T. Wirth, "Transformer-based file fragment type classification for file carving in digital forensics", *European Conference on Cyber Warfare and Security*, vol. 24, no. 1, p. 169, Jun. 2025. doi: 10.34190/eccws.24.1.3552.
- [2] S. Liu, J. G. Park, H. Kim, and J. H. Hong, "A cross-attention multi-scale performer with Gaussian bit-flips for file fragment classification", *IEEE Trans. Inf. Forensics Secur.*, vol. 20, p. 2109, Jan. 2025. doi: 10.1109/tifs.2025.3539527.
- [3] S. Yan, J. Ren, W. Wang, L. Sun, W. Zhang, and Q. Yu, "A survey of adversarial attack and defense methods for malware classification in cyber security", *IEEE Commun. Surv. Tutor.*, vol. 25, no. 1, p. 467, Nov. 2022. doi: 10.1109/comst.2022.3225137.
- [4] K. Aryal, M. Gupta, and M. Abdelsalam, "A survey on adversarial attacks for malware analysis", *arXiv*, Jan. 2021. doi: 10.48550/arxiv.2111.08223.
- [5] Y. Li, M. Cheng, C. Hsieh, and T. C. M. Lee, "A review of adversarial attack and defense for classification methods", *The American Statistician*, vol. 76, no. 4, p. 329, Nov. 2021. doi: 10.1080/00031305.2021.2006781.
- [6] L. Demetrio, B. Biggio, G. Lagorio, F. Roli, and A. Armando, "Functionality-preserving black-box optimization of adversarial Windows malware", *IEEE Trans. Inf. Forensics Secur.*, vol. 16, p. 3469, Jan. 2021. doi: 10.1109/tifs.2021.3082330.
- [7] K. Skračić, J. Petrović, and P. Pale, "ByteRCNN: Enhancing file fragment type identification with recurrent and convolutional neural networks", *IEEE Access*, vol. 11, p. 138176, Jan. 2023. doi: 10.1109/access.2023.3340441.
- [8] Z. Fang, J. Wang, J. Geng, Y. Zhou, and X. Kan, "A3CMal: Generating adversarial samples to force targeted misclassification by reinforcement learning", *Appl. Soft Comput.*, vol. 109, p. 107505, May 2021. doi: 10.1016/j.asoc.2021.107505.
- [9] K. Shaikat, S. Luo, and V. Varadharajan, "A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks", *Eng. Appl. Artif. Intell.*, vol. 116, p. 105461, Oct. 2022. doi: 10.1016/j.engappai.2022.105461.
- [10] D. Li and Q. Li, "Adversarial deep ensemble: Evasion attacks and defenses for malware detection", *IEEE Trans. Inf. Forensics Secur.*, vol. 15, p. 3886, Jan. 2020. doi: 10.1109/tifs.2020.3003571.

- [11] D. Li et al., "A framework for enhancing deep neural networks against adversarial malware", IEEE Trans. Netw. Sci. Eng., vol. 8, no. 1, p. 736, Jan. 2021. doi: 10.1109/tNSE.2021.3051354.
- [12] M. McDaniel and M. Heydari, "Content based file type detection algorithms", Proc. Hawaii Int. Conf. Syst. Sci., Jan. 2003. doi: 10.1109/hicss.2003.1174905.
- [13] G. Mittal, P. Korus, and N. Memon, "FiTy: Large-scale file fragment type identification using convolutional neural networks," IEEE Trans. Inf. Forensics Security, vol. 16, pp. 28–41, 2021, doi: 10.1109/TIFS.2020.3004266.
- [14] M. Felemban, M. Ghaleb, K. M. Saa'im, S. M. Al-Saleh, and A. Almulhem, "File fragment type classification using light-weight convolutional neural networks", IEEE Access, vol. 12, p. 157179, Jan. 2024. doi: 10.1109/access.2024.3486180.
- [15] Y. Wang, W. Liu, K. Wu, K. Yap, and L. Chau, "Intra- and inter-sector contextual information fusion with joint self-attention for file fragment classification", Knowl.-Based Syst., vol. 291, p. 111565, Feb. 2024. doi: 10.1016/j.knsys.2024.111565.
- [16] P. Penrose, R. Macfarlane, and W. J. Buchanan, "Approaches to the classification of high entropy file fragments", Digit. Invest., vol. 10, no. 4, p. 372, Oct. 2013. doi: 10.1016/j.diin.2013.08.004.
- [17] Y. Fratantonio et al., "MAGIKA: AI-powered content-type detection", Proc. Int. Conf. Softw. Eng., p. 2638, Apr. 2025. doi: 10.1109/icse55347.2025.00158.
- [18] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples", arXiv, Dec. 2014. [Online]. Available: <https://arxiv.org/pdf/1412.6572.pdf>.
- [19] B. Kolosnjaji et al., "Adversarial malware binaries: Evading deep learning for malware detection in executables", Proc. Eur. Signal Process. Conf., p. 533, Sep. 2018. doi: 10.23919/eusipco.2018.8553214.
- [20] K. Lucas, S. Pai, W. Lin, L. Bauer, M. Reiter, and M. Sharif, "Adversarial training for raw-binary malware classifiers", arXiv, 2019. [Online]. Available: <https://arxiv.org/abs/1907.07149>.
- [21] F. Kreuk, A. Barak, S. Aviv, M. Baruch, B. Pinkas, and J. Keshet, "Deceiving end-to-end deep learning malware detectors using adversarial examples," arXiv preprint arXiv:1802.04528, 2019.
- [22] L. Kummer, S. Moustafa, N. M. Kriege, and W. N. Gansterer, "Attacking graph neural networks with bit flips: Weisfeiler and Lehman go indifferent", arXiv, Nov. 2023. doi: 10.48550/arxiv.2311.01205.
- [23] R. Taheri, M. Shojafar, F. Arabikhan, and A. Gegov, "Unveiling vulnerabilities in deep learning-based malware detection: Differential privacy driven adversarial attacks", Comput. Secur., vol. 146, p. 104035, Aug. 2024. doi: 10.1016/j.cose.2024.104035.
- [24] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy", arXiv, Jan. 2019. doi: 10.48550/arxiv.1901.08573.
- [25] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy", arXiv, Mar. 2022. doi: 10.48550/arxiv.1805.12152.
- [26] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples", arXiv, May 2017. doi: 10.48550/arxiv.1705.09064.
- [27] A. Goodge, B. Hooi, S. K. Ng, and W. S. Ng, "Robustness of autoencoders for anomaly detection under adversarial impact", Proc. Int. Joint Conf. Artif. Intell. (IJCAI), p. 1244, Jul. 2020. doi: 10.24963/ijcai.2020/173.
- [28] C. Xie, Y. Wu, L. van der Maaten, A. Yuille, and K. He, "Feature denoising for improving adversarial robustness", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), p. 501, Jun. 2019. doi: 10.1109/cvpr.2019.00059.
- [29] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), p. 7132, Jun. 2018. doi: 10.1109/cvpr.2018.00745.
- [30] S. Woo, J. Park, J. Lee, and I. S. Kweon, "CBAM: Convolutional block attention module", in Lect. Notes Comput. Sci., Springer, 2018, p. 3. doi: 10.1007/978-3-030-01234-2_1.
- [31] P. Agrawal, N. S. Punna, S. K. Sonbhadra, and S. Agarwal, "Impact of attention on adversarial robustness of image classification models", in Proc. IEEE Int. Conf. Big Data, Dec. 2021, p. 3013. doi: 10.1109/bigdata52589.2021.9671889.
- [32] D. Zoran, M. Chrzanowski, P.-S. Huang, S. Goyal, A. Mott, and P. Kohl, "Towards robust image classification using sequential attention models", Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), p. 9480, Jun. 2020. doi: 10.1109/cvpr42600.2020.00950.
- [33] H. Yakura, S. Shinozaki, R. Nishimura, Y. Oyama, and J. Sakuma, "Neural malware analysis with attention mechanism", Comput. Secur., vol. 87, p. 101592, Aug. 2019. doi: 10.1016/j.cose.2019.101592.
- [34] W. K. Al-Ghanem, E. U. H. Qazi, T. Zia, M. H. Faheem, M. Imran, and I. Ahmad, "MAD-ANET: Malware detection using attention-based deep neural networks", Comput. Model. Eng. Sci., vol. 143, no. 1, p. 1009, Jan. 2025. doi: 10.32604/cmesci.2025.058352.
- [35] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," in Proc. 4th International Conference on Information Systems Security and Privacy (ICISSP), Portugal, January 2018.
- [36] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection", IEEE Commun. Surv. Tutor., vol. 18, no. 2, p. 1153, Oct. 2015. doi: 10.1109/comst.2015.2494502.
- [37] G. Zhao, P. Liu, K. Sun, Y. Yang, T. Lan, and H. Yang, "Research on data imbalance in intrusion detection using CGAN", PLoS One, vol. 18, no. 10, Art. no. e0291750, 2023. doi: 10.1371/journal.pone.0291750.