drones

MDPI

Article

Dual-Impact Feature Selection for Adversarially Robust, Functionality-Preserving UAV Intrusion Detection

Saleem Alsaraireh [1,*], Mustafa Al-Fayoumi [1,2] and Mohammad Alnabhan [1]

[1] Department of Computer Science, King Hussein School of Computing Sciences, Princess Sumaya University for Technology, Amman 11941, Jordan; m.alfayoumi@psut.edu.jo (M.A.-F.); m.alnabhan@psut.edu.jo (M.A.)
[2] Department of Computer Science, Faculty of Information Technology, Middle East University (MEU), Amman 11831, Jordan
* Correspondence: sal20219008@std.psut.edu.jo

Highlights

What are the main findings?

- The proposed Dual-Impact Feature Selection (DIFS) framework identifies UAV network features critical for both intrusion detection accuracy and cyberattack functionality, enabling the generation of realistic adversarial samples.
- UAV-IDS-CNN models trained with DIFS-based constrained adversarial examples achieve up to 99.8% detection accuracy while reducing Attack Success Rate to as low as 0.002.

What are the implications of the main findings?

- The DIFS framework enables the practical deployment of deep learning-based IDS in high-stakes UAV operations by hardening defenses against sophisticated adversarial attacks.
- UAV security practitioners can focus defensive resources on monitoring the most critical network features, significantly reducing computational overhead on resource-constrained UAV platforms.

Abstract

The increasing deployment of Unmanned Aerial Vehicles (UAVs) in critical operations exposes them to cyberattacks. Although deep learning-based Intrusion Detection Systems (IDSs) are effective, they are susceptible to adversarial attacks that manipulate input features to avoid detection. Conventional feature selection methods do not distinguish between features critical to model accuracy and those essential for preserving cyberattack operational validity. To address this, we propose a Dual-Impact Feature Selection (DIFS) framework for robust UAV-IDS models. Our approach evaluates features based on two criteria: the first is Model Performance Impact (MPI), using Integrated Gradients (IG) and Local Interpretable Model-agnostic Explanations (LIME) to measure feature influence on detection accuracy, and the second is Functionality Preservation Criterion (FPC), a clustering-based method that assesses whether a feature is indispensable for cyberattack execution. Features with high MPI and FPC are identified as Dual-Impact Features (DIFs). We generate constrained adversarial attacks that perturb these DIFs to create realistic evasion samples. Using these samples for adversarial training, we develop three robust UAV-IDS Convolutional Neural Network (CNN) models. Evaluated on three UAV network intrusion datasets, our framework demonstrates improved resilience. The models achieve up to 99.8% detection accuracy while reducing Attack Success Rate (ASR) to as low as

Check for updates

Academic Editors: Lei Shi, Yi Zhou, Xuhui Bu and Lingling Lv

Received: 18 February 2026
Revised: 13 April 2026
Accepted: 30 April 2026
Published: 2 July 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

0.002, supporting their potential for designing adversary-resistant detection systems for UAV networks.

Keywords: UAV security; adversarial attacks; IDS; feature selection; adversarial training; cybersecurity

1. Introduction

Unmanned Aerial Vehicles (UAVs) are connected through ad hoc networks and controlled via smart devices, serving applications from military operations to aerial mapping [1,2]. The network security in these systems is based on security protocols and Intrusion Detection Systems (IDSs) [3,4]. However, every component of a UAV network, including the UAV itself, the Ground Control Station (GCS), and access points, is exposed to cyberattacks such as Denial of Service (DoS), replay, and Man-in-the-Middle (MITM) attacks [1], as illustrated in Figure 1. The growing deployment of UAVs in critical operations has made them targets for sophisticated attacks that exploit weaknesses in communication protocols [5], particularly adversarial attacks that preserve semantic functionality while evading detection [6,7].

A critical challenge in UAV network security is that attackers can manipulate the features of network attacks to evade detection while preserving the operational validity of malicious cyberattacks [7]. For example, an adversary conducting an Evil Twin attack might modify 802.11 frame control flags to avoid triggering anomaly detectors while still successfully impersonating a legitimate access point. Traditional feature selection methods fail to account for this dual objective, leading to IDS models that are either too permissive or prone to false positives under adversarial conditions.

Existing feature selection techniques for IDS, such as Principal Component Analysis (PCA) [8], Random Forest importance ranking [9], etc., focus solely on selecting features that maximize detection performance [10]. They ignore whether features can be perturbed in ways that preserve cyberattack operational validity. For example, selecting the *ip.ttl* feature for Distributed Denial of Service (DDoS) detection might improve accuracy, but adversaries can easily manipulate this feature while maintaining valid packet flows [11]. Similarly, methods such as LIME [12] identify which features are important for classification, but do not consider whether those features are essential for executing cyberattacks.

Moreover, current adversarial training approaches often generate unrealistic samples by perturbing all features indiscriminately. This results in packets that violate protocol specifications (e.g., invalid *wlan.fcs.checksum* values) and would be dropped by real network stacks. Such samples provide limited value for hardening IDS against actual and zero-day threats [13,14].

To summarize, UAV networks are critically vulnerable to sophisticated adversarial attacks, particularly those that craft deceptive yet fully functional (semantic-preserving) network traffic to mimic benign behavior. Current security solutions [13–18] focus mainly on external threats and conventional malware, failing to address the unique challenge of these semantically preserved adversarial attacks that can bypass internal defenses. This drawback, compounded by suboptimal feature selection and high false positive rates in real-time operations, exposes UAV networks to potent threats. Motivated by these critical gaps, this research aims to develop a robust deep learning (DL)-IDS designed to detect semantically preserved adversarial attacks in UAV networks, enhancing their security posture and operational efficacy.

Current approaches to adversarial robustness in UAV-IDS exhibit three critical flaws:

1. Performance-only feature selection: Traditional feature selection techniques such as PCA [8], SHapley Additive exPlanations (SHAP) [19], and LIME [12] focus solely on model performance metrics and ignore functional feature constraints inherent to cyberattack execution.
2. Unconstrained adversarial training: Generated samples often violate network protocol semantics (e.g., impossible packet sizes in DDoS attacks), producing unrealistic examples with limited defensive value.
3. Lack of functionality-constrained adversarial training: Existing methods do not restrict perturbations to features that preserve attack operational validity, producing adversarial samples that would fail in real network environments and thus provide poor coverage of realistic threats.

Research Gap and Proposed Solution: These three limitations reveal a fundamental gap in current UAV-IDS research: existing approaches fail to recognize that adversarially robust intrusion detection requires features that are simultaneously critical for model decisions and essential for cyberattack execution. Moreover, adversarial training must generate samples that preserve network protocol semantics to provide meaningful robustness improvements. To address these gaps, we introduce the Dual-Impact Feature Selection (DIFS) framework, which (1) identifies features with high impact on both model performance (via IG and LIME) and attack functionality (via clustering-based FPC); (2) constrains adversarial perturbations to maintain operational validity within protocol-feasible bounds; (3) enables realistic adversarial training that hardens models against executable threats rather than protocol-violating noise. This approach bridges the disconnect between model-centric robustness and domain-specific attack realism, ensuring that defensive improvements translate to protection against real-world adversarial threats in UAV networks.

To address these gaps, this work makes the following key contributions to the field of adversarial robustness in UAV Intrusion Detection Systems (UAV-IDS):

1. Designing a robust and efficient CNN-1D model (UAV-IDS-CNN-1D) to achieve a strong intrusion detection performance for UAVs in benchmark datasets.
2. A feature-centric adversarial robustness framework that focuses defenses on the subset of features most critical to both the model's decisions and the cyberattack's functionality.
3. Automated FPC Check: Introduced an unsupervised clustering-based method to constrain adversarial perturbations within the functional boundaries of each cyberattack class, ensuring the realism of adversarial examples without requiring domain experts.
4. Explainable AI (XAI)-Driven Robustness: Bridged model performance and adversarial robustness by formally integrating LIME and IG with the FPC for feature selection, representing a dual-impact evaluation approach.
5. Protocol-Aware Adversarial Attacks: Implemented constrained versions of FGSM and PGD attacks that are tailored to the UAV-IDS domain, perturbing only selected features within a semantically feasible space to generate realistic adversarial examples.
6. Comprehensive Robustness Evaluation: Rigorously demonstrated that our approach facilitates significant performance recovery after adversarial retraining, quantitatively hardening the model against realistic, semantically constrained attacks. Performance is evaluated using six complementary metrics: Precision, Recall, F1-Score, and Accuracy quantify detection capability on clean data, while Attack Success Rate (ASR) and Robust Accuracy measure resilience against adversarial perturbations (detailed in Section 5).

The proposed framework reduces the ASR while maintaining high robustness in network traffic. This work establishes a foundation for the development of more resilient cybersecurity systems capable of operating effectively in adversarial environments.

Figure 1 illustrates the three-tier architecture of a typical UAV network and corresponding attack surfaces. The system comprises: (1) the UAV swarm communicating via an ad hoc wireless network, (2) the GCS managing flight operations, (3) three attack vectors. The *Cyberattack-Side* (right) represents network-level threats, including DDoS, Brute-Force, Reconnaissance, and UAV-specific attacks like Fake Landing commands. The *Adversarial-Attack-Side* (left) represents attacks targeting the IDS itself, with evasion attacks manipulating features at inference time, and poisoning attacks corrupting training data. This dual-threat model—conventional cyberattacks against UAV infrastructure and adversarial attacks against the detection system—directly motivates our DIFS framework. The IDS must not only detect diverse cyberattacks but also remain robust against the adversarial manipulation of the features it relies upon.

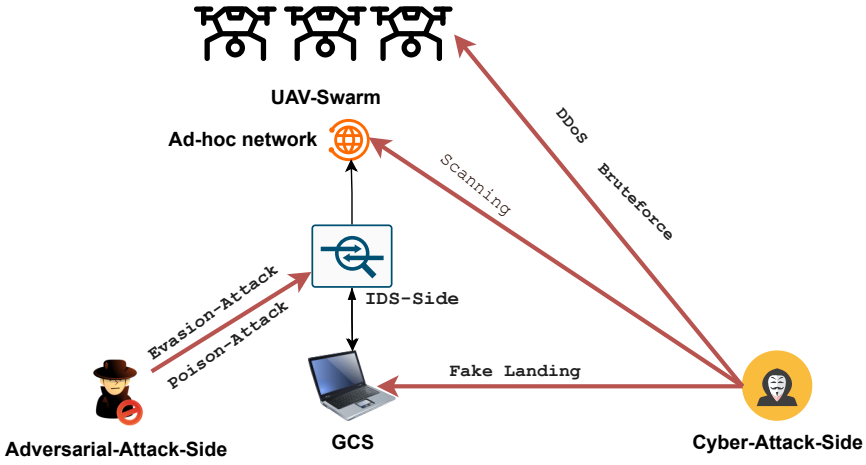


Figure 1. Components of networked UAVs and potential cyberattacks.

The remainder of this paper is organized as follows. Section 2 provides the foundational background on adversarial attacks and their taxonomy. Section 3 presents an overview of related work on UAV-IDS, feature selection, and adversarial defense approaches. Section 4 details the components of our proposed DIFS methodology. The experimental setup and datasets are described in Section 5. Section 6 presents the evaluation results, followed by a discussion of the findings in Section 7. Finally, Section 8 concludes the paper and outlines future research directions.

2. Preliminaries

Combining DL models, primarily CNNs, in UAV applications has achieved notable improvements in IDSs and real-time object detection [20,21]. These models learn hierarchical feature representations from UAV IDS datasets, achieving state-of-the-art performances in classification tasks. Nevertheless, this high performance critically relies on the assumption that the training and test data come from the same distribution. This assumption is often violated in adversarial settings, where models are exposed to gradually manipulated data, revealing a deep exposure to malicious cyberattacks [22]. Security threats to DL models can be taxonomized by the phase of the DL model learning pipeline that is compromised. Figure 2 in the red region shows two primary attack vectors that can affect any DL model: (1) data poisoning, which targets the training process, (2) evasion attacks, which target the model during inference [7,22,23].

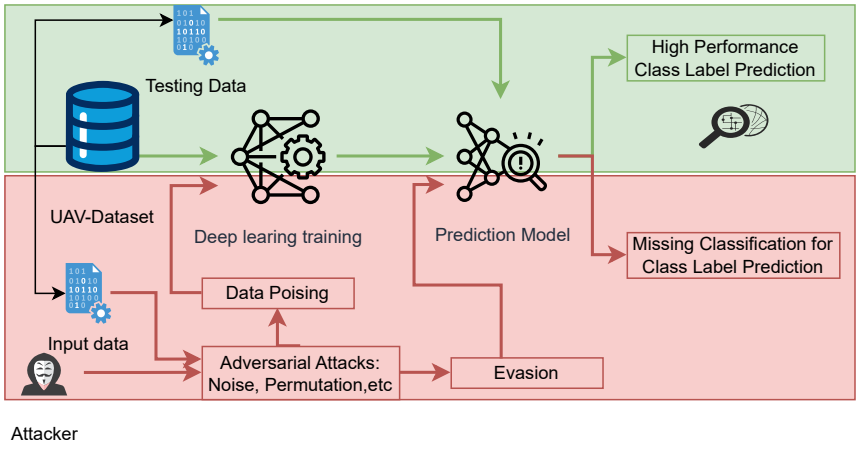


Figure 2. General overview of adversarial attacks on DL model during training and model deployment.

Adversarial attacks represent a critical challenge for machine learning-based security systems. These attacks involve deliberately manipulating input data to cause misclassification, exploiting the mathematical properties of DL models [22,24]. In UAV networks, adversarial attacks pose unique challenges: perturbations must remain within operational protocol constraints to preserve attack operational validity, distinguishing this domain from conventional adversarial machine learning (ML), where semantic preservation is primarily perceptual [7]. The following subsections examine the technical mechanisms of these attacks and their implications for UAV-IDS security.

2.1. Adversarial Attacks

The main adversarial attacks affecting DL models include data poisoning (or training time attacks), which compromise the integrity of the model's learning procedure. An attacker injects a set of adversarial samples D_{poison} into the training dataset D_{train} [24,25]. The model parameters θ are then optimized on the corrupted dataset $D_{\text{train}} \cup D_{\text{poison}}$, leading to a compromised prediction model. As poisoning occurs before deployment, it is difficult to detect; the model appears to perform well until a trigger is activated or its general performance degrades. Formally, the standard training objective is subverted in Equation (1):

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D_{\text{train}}}[J(\theta, x, y)]. \tag{1}$$

This standard training objective optimizes the model parameters θ to minimize the expected loss over the training distribution [23].

The impact of poisoning can be classified into two types. The first is performance degradation, where the model's overall performance on a clean test set D_{test} is reduced. The second is backdoor injection (an integrity attack), where the model performs well on D_{test} but exhibits malicious behavior when a trigger τ is present. The learning objective is covertly altered so that for inputs x from the clean test distribution, we use Equation (2):

$$f(x + \tau; \theta) = y_t \quad \text{while} \quad f(x; \theta) = y \quad \forall (x, y) \in D_{\text{test}}, \tag{2}$$

where y_t is the attacker's target label and τ is the backdoor trigger pattern [23].

Second, evasion attacks [23], often used interchangeably with adversarial attacks, occur after the DL model is trained and deployed. An attacker crafts a perturbation δ to a data input x to generate an adversarial example $x' = x + \delta$ that causes the model to misclassify, i.e., $f(x') \neq y$. The perturbation is constrained by $\|\delta\|_p \leq \epsilon$ to remain small enough to be imperceptible to a human observer. This constraint, typically enforced under

a L_p norm, ensures that the semantic content of x is preserved. The general formulation of an evasion attack can be written as the following constrained optimization problem in Equation (3):

$$\begin{aligned} &\underset{\delta}{\text{maximize}} \quad J(\theta, x+\delta, y) \\ &\text{subject to} \quad \|\delta\|_p \leq \epsilon \quad \text{and} \quad x+\delta \in [0,1]^m, \end{aligned} \tag{3}$$

where ϵ is the maximum perturbation allowed and the input is scaled to $[0,1]^m$. This formulation establishes the core evasion attack framework [22,24].

Several algorithms have been proposed to solve this problem. The Fast Gradient Sign Method (FGSM) [24] is a single-step attack. While computationally efficient, FGSM often leads to suboptimal attacks due to its linear approximation in a highly nonlinear loss landscape. To overcome the limitations of FGSM [26], the Basic Iterative Method (BIM) was introduced, as described in Section 2.3.

2.2. FGSM Adversarial Attack

The FGSM is a simple and efficient one-step attack to generate adversarial examples, illustrating the core vulnerability of models to gradient-based perturbations [24].

Core Idea: Perturb the input data in the direction of the sign of the loss function's gradient with respect to the input. This single step maximizes the loss and leads to misclassification.

$$x_{\text{adv}} = x + \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y)) \tag{4}$$

where

- x_{adv}: The generated adversarial example.
- x: The original clean input.
- ϵ: The perturbation budget, controlling the maximum deviation under the L_∞ norm.
- $\nabla_x J(\theta, x, y)$: The gradient of the model's loss function with respect to the input.

2.3. BIM and Projected Gradient Descent (PGD)

BIM extends FGSM to a more powerful multi-step attack [26]. This approach is formally equivalent to PGD in the negative loss function, and is considered a standard benchmark for evaluating adversarial robustness [27].

Core Idea: Apply FGSM multiple times with a small step size α. After each step, the adversarial example is projected back onto the ϵ-ball around the original input x to ensure that the perturbation constraint is satisfied.

$$x_{\text{adv}}^{t+1} = \text{Proj}_{x+\mathcal{S}}\left(x_{\text{adv}}^{t} + \alpha \cdot \text{sign}(\nabla_x J(\theta, x_{\text{adv}}^{t}, y))\right) \tag{5}$$

where

- x_{adv}^{t}: The adversarial example at iteration t.
- α: The step size, typically set as $\alpha = \epsilon/T$ for T iterations.
- $\text{Proj}_{x+\mathcal{S}}(\cdot)$: The projection operator that confines the adversarial example to the feasible set $\mathcal{S}$, e.g., $\{x' \mid \|x' - x\|_p \leq \epsilon\}$.

As a white-box attack, BIM/PGD is highly potent. The iterative process allows it to navigate the model's nonlinear loss landscape more effectively than FGSM, often finding stronger adversarial examples.

2.4. Comparison and Scope: Poisoning vs. Evasion Attacks

Although both threaten IDSs based on ML, data poisoning and evasion attacks differ fundamentally. Table 1 provides a concise comparison.

Table 1. Comparison of data poisoning and evasion attacks.

Aspect	Data Poisoning	Evasion Attacks
Attack Phase	Training (before deployment)	Inference (after deployment)
Adversary Access	Requires training data access	Requires only query access
Persistence	Permanent until retraining	Transient, sample-specific
Impact Scope	Global (entire distribution)	Local (adversarial samples)
UAV Relevance	Lower (secure training)	Higher (runtime threats)
Defense	Secure pipeline [23]	Robust models [28]

Focus of This Work: Evasion Attacks. This paper addresses evasion attacks, the primary realistic threat to deployed UAV-IDS systems. Our rationale is as follows: (1) operational UAV networks secure training environments, making poisoning difficult; (2) evasion attacks require lower adversary capabilities; (3) deployed systems face continuous evasion attempts [29]. The DIFS framework specifically targets evasion robustness through functionality-preserving adversarial training (Sections 4.3 and 4.4).

3. Related Works

The development of robust IDSs for UAVs sits at the intersection of several critical research areas [1,30]. This section examines the existing literature, starting with intrusion detection in UAV networks, then exploring the feature selection methods that underpin effective IDSs, and finally analyzing the threat of adversarial attacks that can compromise DL-based protection models. This structured review establishes the foundation of this research and identifies the gaps that this research aims to address.

The unique constraints of UAVs, such as limited computational resources and the critical need for real-time operation, have driven research into efficient and accurate IDSs. Previous studies have utilized diverse strategies, often DL models, on benchmark network datasets. For example, Ferrag et al. [30] used multiple DL methods to identify novel attacks and employed 10-fold cross-validation for evaluation. Similarly, Basnet et al. [31] combined the CIC-IDS2017 and CIC-IDS2018 datasets to evaluate the generalizability of an IDS across 12 learning algorithms. Nevertheless, their approach is limited by the assumption that categorical features have the same number of distinct values across both datasets, which may not hold and could compromise the model's integrity.

To address specific attack vectors, such as GPS spoofing, novelty-based detection systems have been proposed. The work in [16] presents a one-class classifier approach trained solely on normal flight logs. By using PCA for dimensionality reduction, their model achieved high F1-scores against GPS spoofing, demonstrating strong cross-platform detection capabilities. Other approaches, such as Binary Decision Trees, have also been reported to achieve near-perfect performances on standard metrics, highlighting the potential of ML for securing UAV networks.

The performance of any ML-based IDS is heavily dependent on the quality and relevance of the input features. Consequently, feature selection (FS) is a critical preprocessing step to reduce computational overhead and improve detection accuracy. Traditional FS methods have focused on statistical properties, using techniques such as Correlation-based Feature Selection (CFS) and information gain [10,32,33].

More recently, advanced optimization and rough set models have been applied to this problem. For example, Han et al. [34] presented a binary Particle Swarm Optimization (PSO) method with a wrapper-based approach to improve the interaction between feature selection and model training. To further enhance performance, Almomani [33] introduced an ensemble FS model that combines the gray wolf optimizer, genetic algorithms, the firefly algorithm and PSO. In the context of UAVs, a lightweight IDS was proposed in [35], leveraging an improved fuzzy rough set model with adaptive neighborhoods for efficient feature selection that handles partially missing data.

XAI has also emerged as a powerful tool for feature selection. As demonstrated in [36], SHAP can identify the most significant features in datasets such as CICIDS2017. By creating optimal feature subsets, they showed that a reduced feature set can maintain a high accuracy for models such as CNNs and Random Forest (RF). This aligns with frameworks like Enhanced Dynamic Filter-Based Feature Selection (E-DFBFS), which uses NSGA-II to achieve a superior accuracy and F-measure.

While DL models have proven highly effective, they are notoriously vulnerable to adversarial attacks—carefully crafted inputs designed to cause misclassification. Although early investigations focused on the image domain [20,21,37], this threat has since been extended to network security, including IDSs.

Research by Yang et al. [38] confirmed this threat to DL-based network IDSs, using model substitution to execute successful black-box attacks. Subsequent studies have further explored these vulnerabilities. For instance, Wang et al. [39] analyzed the divergence between the manifold of adversarial instances and the model's decision boundaries, while Peng et al. [40] systematically benchmarked the performance of various ML models under adversarial settings. In response, defense mechanisms are being developed. Debicha et al. [41] analyzed the impact of attacks and presented adversarial training as a defensive countermeasure. Sauka et al. [28] merged adversarial training with an XAI framework based on SHAP to address both vulnerability and model interpretability. The threat to Internet of Things (IoT) ecosystems is also being investigated, with Papadopoulos et al. [42] studying the impact of adversarial attacks on ML-based IDS in IoT.

Despite these advances, a notable study gap continues in the design of adversarially robust IDSs for UAV networks. First, many investigations focus on assessing model performance against known attacks but fail to proactively build robust DL models that can resist adversarial perturbations. Second, while feature selection methods prioritize metrics that improve detection accuracy, they often neglect to constrain the feature set to those that preserve the operational semantics of cyberattacks. This direction means that adversarial instances, though effective in misleading the model, may render non-functional or malformed network traffic that would not be viable in a real-world cyberattack scenario. Finally, the prevailing adversarial training methods do not focus on training models, especially on a refined subset of high-impact, functionality-preserving features, thereby restricting their efficacy in the resource-constrained UAV environment.

Table 2 summarizes the key differences between existing UAV-IDS and adversarial robustness approaches and our proposed DIFS framework, highlighting the novel contributions of this work.

Table 2. Comparison of existing approaches with the proposed DIFS framework.

Approach	Feature Selection	Adversarial Robustness	Functionality Preservation	XAI Integration	UAV-Specific
Ferrag et al. [30]	Standard (all features)	Not addressed	N/A	No	Yes
Basnet et al. [31]	Standard	Not addressed	N/A	No	No
Yang et al. [38]	Standard	Attack evaluation only	N/A	No	No
Sauka et al. [28]	SHAP-based	Adversarial training	No	Yes (SHAP)	No
Debicha et al. [41]	Standard	Adversarial training	No	No	No
Han et al. [34]	PSO optimization	Not addressed	N/A	No	No
Almomani [33]	Ensemble (GA + PSO + FF)	Not addressed	N/A	No	No
Chen et al. [36]	SHAP-based	Not addressed	N/A	Yes (SHAP)	No
This Work (DIFS)	Dual-impact (MPI + FPC)	Constrained adversarial training	Yes (FPC-based)	Yes (IG + LIME)	Yes

4. Methodology

This section describes the pipeline of the proposed approach, which focuses on improving the adversarial robustness of DL models within a UAV-IDS. The method is structured into four sequential phases, each building on the previous to achieve robustness improvements.

The process starts with the structure and initial evaluation of three CNN-1D-based UAV-IDS models, as described in Section 4.2. Following this foundational step, we use our feature-scoring method (DIFS) in Section 4.3. The output of this feature selection phase is necessary for the following evaluation, facilitating a focused examination of CNN models against C-FGSM and C-BIM constrained adversarial attacks in Section 4.4. In Section 4.5, the insights achieved from this adversarial evaluation directly inform the final phase: the implementation of an adversarial retraining strategy to construct a robust and interpretable final detection model. Figure 3 gives an overview of the overall proposed methodology.

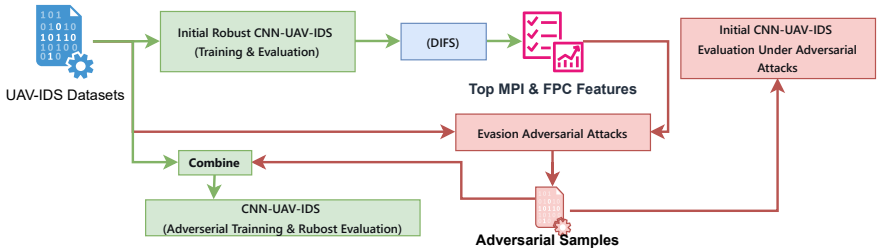


Figure 3. Overview of the proposed DIFS framework for adversarially robust UAV IDS.

4.1. Terminology

Throughout this paper, we use the following terminology consistently:

- Robustness refers to the model's ability to maintain a high detection accuracy under adversarial perturbations. We quantify robustness using Robust Accuracy and ASR. A robust model exhibits high Robust Accuracy and low ASR.
- Resilience describes the model's capacity to recover and maintain performance after adversarial retraining. Resilience is demonstrated through improvements in ASR reduction when comparing initial to retrained model performance. A resilient model shows significant performance recovery after exposure to adversarial examples.

- Functionality-preserving pertains to adversarial samples that maintain the operational semantics of the original cyberattack. We ensure this property through the FPC, which identifies features whose alteration would render the cyberattack non-functional.

4.2. Baseline Deep UAV-CNN-1D Training

This section presents, step by step, the design and training of the best CNN-1D model architecture. The model was trained using the three UAV-IDS datasets to create three UAV-IDS-CNN-1D models. The objective of this step is to establish a high-performance benchmark model on unperturbed data, which will serve as the foundation for robustness analysis and the feature selection approach in subsequent models.

4.2.1. Dataset Preprocessing

This paper utilizes three UAV-IDS datasets, which are described in detail in Section 5.1, containing labeled instances of normal traffic and a diverse set of 10 attack categories: Scanning, Reconnaissance, DoS, DDoS, GPS Jamming, MITM, Replay, Evil Twin, Brute-Force, and Fake Landing. We use an advanced preprocessing pipeline that includes the following standard formal steps.

1. Data Cleaning: Handle missing values using robust imputation (median for numerical features, mode for categorical features).
2. Categorical Encoding: Convert all categorical features (e.g., protocol type) into numerical representations using label encoding.
3. Feature Scaling: Standardize all numerical features using *StandardScaler* to ensure stable and efficient convergence of the model.

After preparing the UAV-IDS datasets for the evaluation of the main model, we use stratified data splitting, where data are partitioned into training sets (70%), validation sets (15%), and test sets (15%) using stratified sampling. This guarantees that the distribution of all attack classes is preserved across splits, ensuring a fair and representative evaluation. Figure 4 illustrates the training and evaluation pipeline of the UAV-IDS-CNN-1D model.

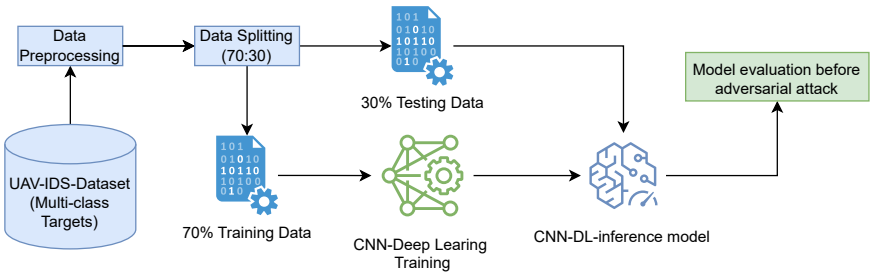


Figure 4. Phase 1: Initial UAV-IDS-CNN-1D training and testing.

4.2.2. UAV-IDS-CNN-1D Model Architecture

Rationale for 1D-CNN Architecture: We select CNN-1D based on domain-specific considerations for UAV network intrusion detection. CNN-1D architectures extract local patterns from sequential packet-level features through 1D convolution kernels [43], enabling parallel processing for faster inference compared to recurrent architectures (LSTM, GRU), which is critical for resource-constrained UAV platforms [44]. Prior work has demonstrated CNN-1D's strong performance on network intrusion detection tasks [45], making it a proven baseline for adversarial robustness evaluation. The core of our IDS is a CNN-1D designed to hierarchically extract features from sequential network traffic data. The primary objective of this phase is to construct and evaluate the CNN UAV-IDS model before generating adversarial attacks. During this phase, we train three CNN models using

different UAV-IDS datasets, as each dataset is constructed under different scenarios. The architecture, summarized in Figure 5, is as follows:

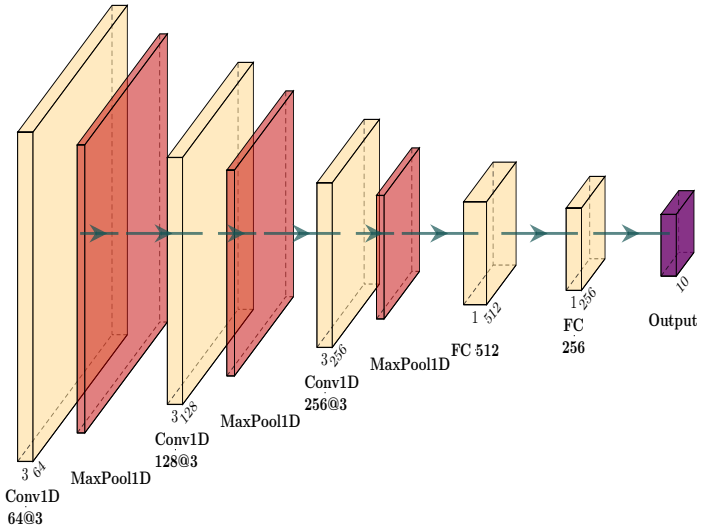


Figure 5. CNN-1D model architecture summary.

1. Input and Convolutional Blocks: The network accepts a 1D vector of preprocessed features. This is processed by three convolutional blocks. Each block contains a 1D convolutional layer with ReLU activation, batch normalization, and a max-pooling layer (pool size = 2) for progressive feature extraction and dimensionality reduction. The number of filters increases block by block (64, 128, and 256) to capture more complex patterns.
2. Classification Head: The output of the final convolutional block is flattened and fed into two dense layers (512 and 256 units, each with ReLU activation). A dropout rate of 0.3 is applied after each dense layer to mitigate overfitting. The final output layer is a dense layer with a softmax activation function, corresponding to the number of classes (10 attacks + normal).
3. Training Configuration: To handle the inherent class imbalance in the dataset, we used a weighted cross-entropy loss function, where class weights were calculated inversely proportional to their frequencies. The model was trained with the Adam optimizer using an initial learning rate of 0.001. We further incorporated two regularization techniques: a `ReduceLROnPlateau` scheduler to adapt the learning rate upon validation loss stagnation and early stopping with a patience of 10 epochs to restore the best model weights and prevent overfitting.

4.3. DIFS Framework

This phase focuses on selecting the most critical DIFs to build a robust UAV-IDS-CNN-1D model. In contrast to existing methods that prioritize only a model's predictive performance, our approach identifies the top ten features that are both significant for the model's predictions and semantically essential to the operational validity of cyberattacks. This dual focus on performance and functional semantics is fundamental for improving the model's resilience against realistic adversarial attacks. We employ two complementary feature importance techniques to quantitatively assess feature contributions. The overall DIFS process is illustrated in Figure 6, while Algorithm 1 provides a detailed, step-by-step explanation with illustrative examples.

MPI Calculation: We first use IG to compute the integral of gradients along the path from a baseline input to the actual input, providing a principled and efficient mechanism for

global feature attribution. Subsequently, we employ LIME on specific, difficult-to-classify samples to validate that the features identified as significant in a global context remain relevant locally. IG primary attribution scores are normalized and aggregated into a global MPI score, allowing the identification of the top features consistently leveraged by the model. The MPI score for feature i is mathematically represented in Equation (6):

$$\text{MPI}_i = \frac{1}{N} \sum_{j=1}^{N} |\text{Attr}_i(x_j)| \tag{6}$$

where

- N denotes the total number of instances analyzed.
- $\text{Attr}_i(x_j)$ represents the attribution score for feature i corresponding to instance j.

FPC Calculation: To automatically identify the most critical features for a cyberattack's functionality, we introduce the FPC score. This approach begins by modeling the natural variations within a cyberattack class using an unsupervised cluster-based functional representation. For each cyberattack class, we apply K-means clustering to its training samples, with the optimal number of clusters determined using the silhouette score. The core hypothesis is that each cluster represents a distinct semantically valid cyberattack.

This clustering enables a cluster-constrained perturbation strategy. Instead of random permutation, we perturb a feature by shifting its value toward the centroid of its own cluster or a nearby cluster of the same attack class. This ensures that the perturbed sample remains a realistic and functionally consistent sample of the original cyberattack.

The FPC score for a given feature is then calculated as the performance drop (e.g., a decrease in F1-Score) when the model is evaluated on a test set in which that feature has been perturbed using this method. A large performance drop indicates that the feature is as functionally critical as its perturbation, even within semantically valid bounds. We then select the top 10 features with the highest FPC scores as those most essential for preserving the semantic meaning of the cyberattack after any permutations. FPC measures the performance impact of the feature perturbation calculated in Equation (7):

$$\text{FPC}_i = \text{F1}_{\text{original}} - \text{F1}_{\text{perturbed}(i)} \tag{7}$$

In this work, the FPC metric is introduced as a novel approach to quantifying feature criticality from a perspective of cyberattack functionality, complementing existing model-centric feature importance methods [12,19].

The final feature ranking, which combines MPI and FPC, is calculated using Equation (8). The resulting features are classified into four quadrants based on their scores, as defined in Table 3.

$$\text{Score}_i = \alpha \cdot \text{MPI}_{\text{norm},i} + (1 - \alpha) \cdot \text{FPC}_{\text{norm},i} \tag{8}$$

Table 3. DIFS quadrant classification.

	High FPC	**Low FPC**
High MPI	Core Features (Keep)	Vulnerable Features (Harden)
Low MPI	Ancillary Features (Conditional Keep)	Noise Features (Drop)

The methodology of the DIFS framework ensures that selected features cannot be manipulated by adversaries without fundamentally altering the nature of cyberattacks, providing inherent robustness against evasion attempts while maintaining a high detection accuracy.

4.3.1. Rationale for Dual-Criteria Feature Selection

The DIFS framework deliberately combines two complementary criteria, model MPI and FPC, because neither criterion alone is sufficient for generating realistic, adversarially robust samples.

Why MPI alone is insufficient: MPI (via IG and LIME) identifies features the model relies upon most heavily for classification. However, a high-MPI feature may be trivially manipulable by an adversary without affecting the attack's real-world behavior. For example, the `ip.ttl` field is highly informative for DDoS detection (high MPI), yet an attacker can freely modify TTL values while maintaining a valid packet flow. Perturbing such features generates adversarial samples that evade detection but no longer represent functional cyberattacks.

Why FPC alone is insufficient: FPC identifies features whose modification fundamentally breaks attack functionality. However, a high-FPC feature may contribute little to the model's classification decision. Focusing only on FPC could select features that are operationally critical but provide no adversarial leverage—yielding weak adversarial examples that fail to challenge the model.

Are MPI and FPC independent? The two criteria are partially correlated but measure distinct properties. MPI reflects the model's internal decision landscape, while FPC reflects the operational structure of network cyberattacks. A feature can have high MPI and low FPC (e.g., `ip.ttl`—important to the model, easy to perturb without breaking the attack), low MPI and high FPC (e.g., `wlan.fc.subtype`—critical to attack execution but underweighted by the model), or high MPI and high FPC (e.g., `radiotap.channel.flags.cck`—both model-critical and operationally essential). Table 3 captures this four-quadrant structure. Only features in the top-left quadrant (Core Features) are selected as DIFs, indicating that both criteria are jointly necessary.

Real-network example: Consider the `wlan.fc.subtype` feature in a de-authentication attack. This field carries the management frame subtype, where subtype = 12 indicates a de-authentication frame. From an FPC perspective, any perturbation away from 12 fundamentally transforms the packet from a de-authentication attack into normal data traffic, thereby breaking the attack's functionality entirely. From an MPI perspective, IG assigns a high attribution score to this feature because the model has learned its discriminative value. This dual criticality makes `wlan.fc.subtype` an ideal DIF: perturbing it within functional bounds forces the model to learn robust representations, while ensuring the generated adversarial example remains a valid cyberattack.

4.3.2. DIFS: Illustrative Examples with Algorithm 1

To solidify the conceptual understanding of DIFS from the previous diagrams and equations, we now present a detailed numerical example. We will walk through each key step of Algorithm 1 with a sample feature of cyberattacks, demonstrating how the rankings are computed in practice. The results of this walkthrough are consolidated in Table 4.

Lines 3–6: MPI Analysis Example: The algorithm computes MPI scores by evaluating each feature's importance through IG and LIME validation. For instance, when analyzing network traffic features, `udp.length` might receive a high MPI score of 0.156. This occurs because the model identifies that larger packet sizes (e.g., >1450 bytes) are strong indicators of flooding attacks, whereas normal traffic typically exhibits smaller packets (58–1518 bytes). The LIME validation step confirms that this feature's importance remains consistent even in borderline cases, ensuring that global importance translates to local relevance.

Algorithm 1 DIFS

Require: Feature set F, trained model M, dataset D, clusters k, weight α
Ensure: Selected feature subset S
Initialize $MPI_scores \leftarrow \{\}$, $FPC_scores \leftarrow \{\}$
$F1_{original} \leftarrow \text{Evaluate}(M, D)$
for each feature $f_i \in F$ **do** ▷ Lines 3–6: MPI Analysis
  $IG_scores \leftarrow \text{ComputeIG}(M, D, f_i)$
  $LIME_scores \leftarrow \text{ComputeLIME}(M, D, f_i)$
  $MPI_scores[f_i] \leftarrow \text{CombineAttributions}(IG_scores, LIME_scores)$
end for
$centroids \leftarrow \text{KMeansClustering}(D, k)$ ▷ Lines 8–17: FPC Analysis
for each feature $f_i \in F$ **do**
  $D_{perturbed} \leftarrow \text{Copy}(D)$
  for each sample $x \in D_{perturbed}$ **do**
    $cluster_id \leftarrow \text{AssignCluster}(x, centroids)$
    $x[f_i] \leftarrow centroids[cluster_id][f_i]$
  end for
  $F1_{perturbed} \leftarrow \text{Evaluate}(M, D_{perturbed})$
  $FPC_scores[f_i] \leftarrow F1_{original} - F1_{perturbed}$
end for
$MPI_{norm} \leftarrow \text{Normalize}(MPI_scores)$ ▷ Lines 19–22: Score Combination
$FPC_{norm} \leftarrow \text{Normalize}(FPC_scores)$
$combined_scores \leftarrow \alpha \cdot MPI_{norm} + (1 - \alpha) \cdot FPC_{norm}$
$S \leftarrow \text{TopKFeatures}(combined_scores, k = 10)$
return S

Lines 8–17: FPC Analysis Example: The algorithm performs feature perturbation clustering by first grouping the dataset into k clusters (e.g., $k = 5$) representing different traffic patterns. The clustering reveals distinct behavioral patterns:

- Cluster 1 (Normal traffic): Characterized by small packet sizes: `frame.len`: 84B, `udp.length`: 58B.
- Cluster 2 (Flooding attacks): Exhibits large packet sizes: `frame.len`: 1512B, `udp.length`: 1450B.
- Cluster 3 (Wireless attacks): Shows specific wireless signatures: `wlan.fc.subtype`: 8, `radiotap.dbm_antsignal`: −52 dBm.

For each feature, values are systematically replaced with their corresponding cluster centroid values. For example, when perturbing `udp.length`, a sample originally containing `udp.length: 1450B` from Cluster 2 maintains this value after perturbation since it matches the centroid. This approach preserves data distribution while testing feature importance.

The FPC score is calculated as performance degradation after perturbation. Example results demonstrate varying impact levels:

- `udp.length`: FPC = 0.892 − 0.469 = 0.423 (critical impact).
- `wlan.fc.subtype`: FPC = 0.892 − 0.494 = 0.398 (critical impact).
- `frame.len`: FPC = 0.892 − 0.521 = 0.371 (high impact).

Semantic Impact Analysis: The perturbation reveals critical semantic changes. For de-authentication attacks, the original feature `wlan.fc.subtype` equals 12 (indicating de-auth frame), leading to correct attack detection. However, when perturbed to different values such as 0 (data frame), the attack becomes undetectable, demonstrating how fundamental frame-type alterations can transform malicious traffic into normal behavior.

Lines 19–22: Score Integration Example: The algorithm normalizes both MPI and FPC scores and combines them using the weight parameter α. The resulting rankings reveal complementary strengths.

Table 4. Example feature ranking by DIFS.

Feature	MPI Score	FPC Score	Combined Score
`frame.len`	1.000	0.875	0.938
`udp.length`	0.769	1.000	0.884
`wlan_radio.signal_dbm`	0.890	0.734	0.812
`wlan.fc.subtype`	0.538	0.939	0.739
`ip.ttl`	0.538	0.839	0.689

4.4. Performing Adversarial Attacks Using DIFS

This subsection details the application of adversarial attacks, specifically FGSM and BIM, constrained to use only the features identified by DIFS lists for each UAV-IDS dataset. This methodology ensures the generation of functional adversarial samples that preserve the semantic integrity of the original cyberattacks.

These crafted samples serve a two-fold evaluation: first, to assess the adversarial ASR against the initially deployed UAV-IDS-CNN-1D model; and second, to augment the training data for retraining and re-evaluating the robustness of the three original UAV-IDS-CNN-1D models. Figure 6 illustrates the overall adversarial evaluation pipeline. The process begins with selecting the top 10 DIFs. These 10 DIFs are then used to generate adversarial perturbations via FGSM or BIM. Finally, the resulting adversarial samples are used to rigorously evaluate the robustness of the initial UAV-IDS-CNN-1D models.

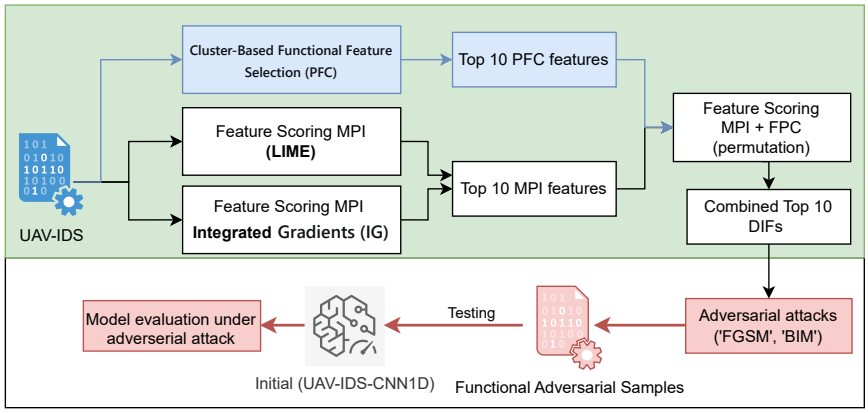


Figure 6. The DIFS framework and evaluation pipeline. Phase 1 (green): DIFS process for selecting robust features. Phase 2 (white): UAV-IDS-CNN-1D model evaluation under functional adversarial attacks.

Constrained Adversarial Attack Formulation

While Section 1 described standard FGSM and BIM attacks, our implementation applies Constrained FGSM (C-FGSM) and Constrained BIM (C-BIM), specifically designed for functionality-preserving adversarial generation.

Constraint 1: Feature Space Restriction

Perturbations are applied only to the top-10 DIFs identified by the DIFS framework (Section 4.3). For a sample $\mathbf{x} \in \mathbb{R}^n$ with n total features, let $\mathcal{D} \subset \{1, 2, \ldots, n\}$ denote the set of DIF indices (with $|\mathcal{D}| = 10$), as formalized in Equation (9):

$$\mathbf{x}_{\text{adv}}[i] = \begin{cases} \mathbf{x}[i] + \delta[i], & \text{if } i \in \mathcal{D} \\ \mathbf{x}[i], & \text{otherwise} \end{cases} \tag{9}$$

where $\delta[i]$ is computed using FGSM (Equation (4)) or BIM (Equation (5)) for feature $i \in \mathcal{D}$. This reduces the adversarial search space from $\mathbb{R}^n$ to $\mathbb{R}^{10}$, focusing on features critical for both model decisions and attack functionality.

Constraint 2: Functionality Preservation Bounds

Standard adversarial attacks use uniform ϵ-ball constraints that ignore domain-specific validity ranges. We derive functionality-preserving bounds from FPC clustering analysis. For each cyberattack class c and each DIF i, perturbations are constrained as defined in Equation (10):

$$\mathbf{x}_{\text{adv}}[i] \in [\mu_c[i] - \sigma_c[i], \mu_c[i] + \sigma_c[i]] \tag{10}$$

where $\mu_c[i]$ and $\sigma_c[i]$ are the mean and standard deviation of the feature i within the attack class c's cluster. This ensures that perturbed values remain within empirically observed ranges of functional attacks. For example, if de-authentication attacks exhibit `wlan.fc.subtype` values consistently near 12, perturbations are constrained to this range, preventing protocol-violating frames.

The projection operator in C-BIM is modified accordingly, as shown in Equation (11):

$$\text{Proj}_{\text{FPC}}(\mathbf{x}'[i]) = \text{clip}(\mathbf{x}'[i], \mu_c[i] - \sigma_c[i], \mu_c[i] + \sigma_c[i]) \tag{11}$$

The selection of features $|\mathcal{D}| = 10$ balances two practical considerations: sufficient coverage of the most critical feature relevant to the model and feature relevant to attack, as determined by the DIFS ranking, while maintaining a tractable perturbation space that preserves interpretability for security professionals.

These constraints ensure C-FGSM and C-BIM generate adversarial samples that (1) maintain operational validity as cyberattacks, (2) reflect realistic evasion attempts adversaries could deploy, (3) focus on the most consequential feature perturbations identified by DIFS.

4.5. UAV-IDS-CNN-1D Adversarial Retraining Phase

This subsection presents the final phase of our approach: a hardening framework for the initially proposed UAV-IDS-CNN-1D model. The process begins by combining the functionally crafted adversarial samples with the original clean training data. This step is crucial for two reasons: it serves as data augmentation and, more importantly, enhances the model's generalization against realistic adversarial examples.

Subsequently, the combined dataset is split into training (70%), validation (15%), and test (15%) sets, consistent with the original partitioning described in Section 4.2.1. The model then undergoes 10 epochs of retraining to refine its defenses. Finally, the robustness of the hardened UAV-IDS-CNN-1D model is rigorously re-evaluated against both adversarial and clean test samples. The objective of this phase is to strengthen existing models, adopt a zero-trust posture, and significantly reduce the effectiveness of potential adversarial attacks. Figure 7 depicts the complete adversarial training pipeline.

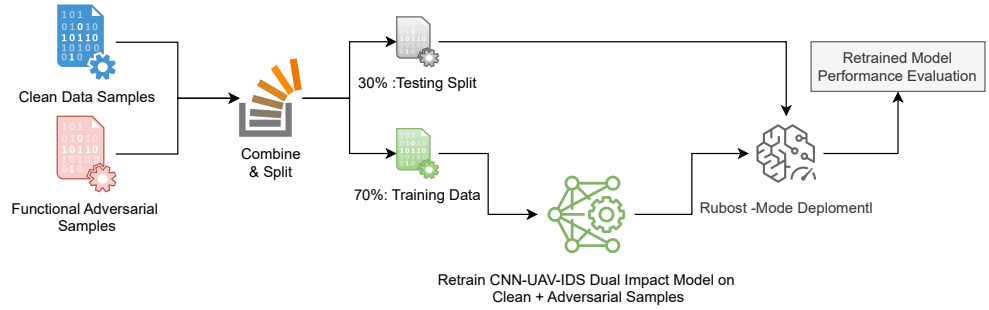


Figure 7. Adversarial retraining and evaluation pipeline for UAV-IDS-CNN-1D. Phase 4 demonstrates the defensive retraining process using adversarial examples.

5. Experimental Validation

5.1. Datasets and Setup

The UAV-NIDD (Unmanned Aerial Vehicle Network Intrusion Detection Dataset) is a real-time dataset designed for cybersecurity intrusion detection in UAV networks. It was generated using a fully functional real UAV testbed. The dataset simulates cyber–physical interactions under benign and malicious network conditions. Unlike earlier synthetic or simulation-based datasets [29], UAV-NIDD captures real traffic from UAVs, GCS, and Access Points (APs) using actual devices and protocols, such as MAVLink and the DJI SDK, making it directly functional for IDSs in UAV networks. The UAV-NIDD dataset is divided into three main dataset scenarios, each emulating a distinct compromise point in the UAV network infrastructure [29].

These scenarios generate three distinct UAV-NIDD datasets developed using a physical testbed involving three UAVs, two GCSs, and dedicated attack and sniffing tools. Table 5 describes the generation of scenarios and the number of features for each dataset, whereas Table 6 details the distribution of the cyberattack classes within them.

Table 5. Dataset generation scenario and number of features.

Dataset	Description	No. of Features
Cyberattack on UAV	Compromised UAV (DJI Mavic Air) initiates DoS/DDoS attacks	45
Cyberattack on Access Point	Rogue AP (via Evil Twin or MITM) hijacks communication	51
Cyberattack on GCS	GCS (Laptop-1) issues malicious control like Fake Landing commands	85

Table 6. Distribution of cyberattack classes across UAV scenarios.

Attack Type	Scenario 1 (UAV)	Scenario 2 (AP)	Scenario 3 (GCS)
Network Attacks			
DDoS	193,171	31,591	93,862
DoS	3213	23,612	206,681
UDP Flooding	152,775	–	–
ICMP Flooding	47,544	–	–
De-authentication	14,606	155,488	1321
Access Attacks			
Brute-Force	175,925	172,509	10,161
Evil Twin	–	10,034	59
Reconnaissance			
Scanning	9030	70,319	44,464
Reconnaissance	6	20	14,285
Specialized UAV Attacks			
GPS Jamming	41,600	–	–
Fake Landing	24,050	24,584	489
MITM	151,586	5016	2472
Replay	25,977	8525	–
Normal Traffic	7923	129,541	97

Note: '–' indicates attack type not present in scenario.

This dataset, prepared for real UAV networks, provides a comprehensive view of the interactions among UAV network components, including UAVs, GCS, and AP. It captures multilayered relationships and attack characteristics across various cyber threats, making it well suited for investigating adversarial robustness while considering the functional properties of network features.

5.2. Evaluation Metrics

We evaluate DIFS using both standard classification metrics and adversarial robustness measures, summarized in Table 7. Standard metrics—Precision, Recall, F1-Score, and Accuracy—follow established IDS evaluation protocols [30]. Precision is critical to minimize false alarms that would unnecessarily ground missions or trigger expensive incident response [46], while Recall ensures comprehensive threat detection with no malicious traffic evading the system [5]. F1-Score balances these competing objectives and remains robust to the class imbalance typical of UAV networks, where normal traffic vastly outnumbers specific attack classes [30].

For adversarial evaluation, we adopt the Attack Success Rate (ASR) and Robust Accuracy following robustness assessment frameworks [27,28]. ASR quantifies the percentage of correctly classified samples that become misclassified under adversarial perturbation, while Robust Accuracy ($1 - \text{ASR}$) directly measures maintained detection capability under attack conditions.

Table 7. Classification and robustness evaluation metrics.

Metric	Formula	Description
Accuracy	$\frac{TP+TN}{TP+TN+FP+FN}$	Ratio of correct predictions to total samples.
Precision	$\frac{TP}{TP+FP}$	Correct positive predictions among all predicted positives.
Recall	$\frac{TP}{TP+FN}$	Actual positives correctly identified by the model.
F1-Score	$2 \cdot \frac{Precision \cdot Recall}{Precision+Recall}$	Harmonic mean of Precision and Recall.
ASR	$\frac{N_{\text{misclassified}}}{N_{\text{correct}}}$	Fraction of correct clean predictions that become misclassified under adversarial perturbation.
Robust Acc.	$1 - \text{ASR}$	Maintained accuracy under adversarial attack conditions.

5.3. Experimental Configuration

All experiments were conducted using Python 3.8, PyTorch 1.9.0, and scikit-learn 1.0.2. We used NVIDIA RTX 3090 GPUs for accelerated training. The specific hyperparameters are detailed in Table 8. We employ a stratified 70:30 train–test split with three random seeds to ensure statistical significance. All results report mean values with standard deviations for multiple runs.

Table 8. Hyperparameters for UAV-IDS-CNN-1D architecture and training procedure.

Parameter	Value
Learning rate	0.001
Batch size	2048
Dropout rate	0.3
Epochs (initial training)	20
Epochs (adversarial retraining)	10
Patience (early stopping)	5
Adversarial perturbation bound (ϵ)	0.03

6. Results and Analysis

This section presents a comprehensive performance evaluation of the proposed UAV-IDS-CNN-1D models. The analysis is structured in four progressive stages to thoroughly assess the system's capabilities. First, baseline performance is established by evaluating the models on clean, unaltered test data without adversarial attacks. Subsequently, the models' inherent vulnerability is assessed by subjecting them to adversarial evasion attacks using the full feature set. Following this, a DIF analysis is performed to identify a robust subset of features that are critical to both high classification accuracy and the preservation of the semantic integrity of cyberattacks. Finally, the robustness of the UAV-IDS-CNN-1D models is re-evaluated under the same evasion attacks using the selected features from the DIFS analysis to demonstrate the effectiveness of the proposed feature selection method.

6.1. Phase 1: Baseline UAV-IDS-CNN-1D Performance Results

To establish a baseline for evaluating robustness, this section first presents the performance of the proposed CNN-1D architecture on clean test data. These initial results are a prerequisite for the subsequent adversarial robustness analysis.

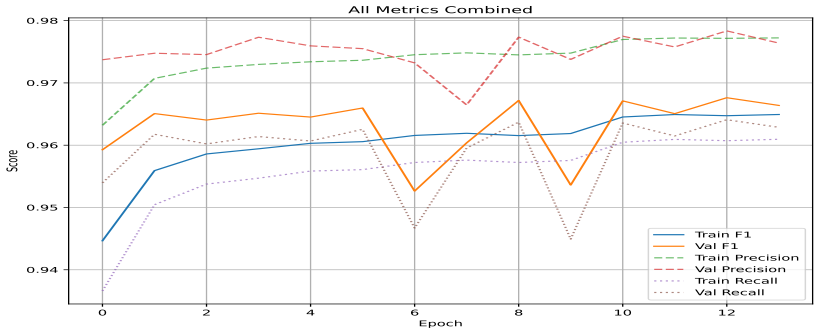
The classification reports for the three models are detailed in Tables 9–11. Table 9 presents the initial performance of the UAV1-IDS-CNN-1D model, demonstrating strong detection capabilities for most attack categories, with a high performance for classes such as Brute-Force, DDoS, and Normal. Nevertheless, it shows challenges in detecting Reconnaissance attacks and a lower performance on De-authentication and Fake Landing.

For comparative analysis, Table 10 shows the results for the UAV2-IDS-CNN-1D model, which achieves a high performance in nearly all classes, with a slight decrease in performance for DDoS attacks. Table 11 reveals a more varied performance profile for the UAV3-IDS-CNN-1D model, with excellent results for Brute-Force and DDoS, but significant difficulties in correctly classifying Benign traffic, De-authentication, Evil Twin, and Fake Landing attacks.

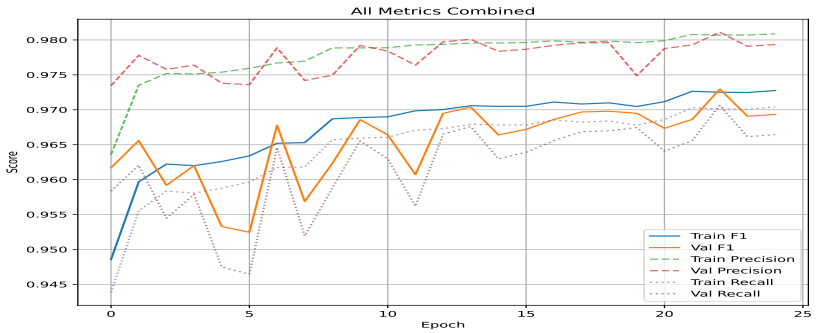
Moreover, the training progression of the three models is illustrated in Figure 8. The sub-figures plot Precision, Recall, and F1-score over 25 training epochs, highlighting the epoch at which early stopping was triggered to prevent overfitting.

Table 9. Classification results of the CNN-1D model on the UAV1-IDS dataset.

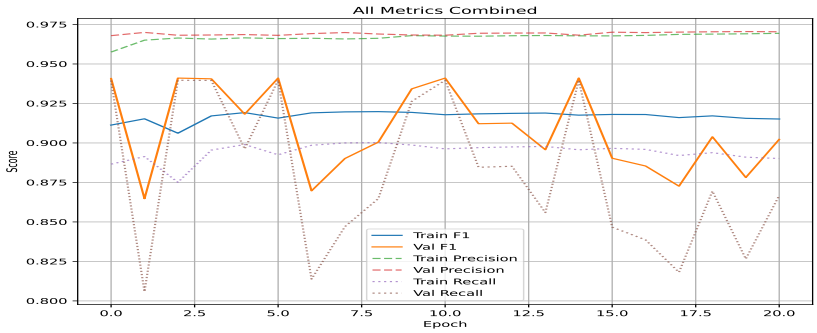
Class	Precision	Recall	F1-Score	Support
Brute-Force	1.000	1.000	1.000	35,185
DDoS	1.000	1.000	1.000	38,634
De-authentication	0.880	0.690	0.770	2921
DoS	0.870	0.980	0.920	643
Fake Landing	0.470	1.000	0.640	4810
ICMP Flooding	1.000	1.000	1.000	9509
Jamming	1.000	1.000	1.000	8320
MITM	1.000	0.820	0.900	30,317
Normal	1.000	1.000	1.000	1585
Reconnaissance	0.000	0.000	0.000	1
Scanning	1.000	1.000	1.000	1806
UDP Flooding	1.000	1.000	1.000	30,555
Replay	0.840	0.950	0.890	5196
Accuracy			0.960	169,482
Macro Avg	0.850	0.880	0.860	169,482
Weighted Avg	0.980	0.960	0.960	169,482



(**a**) UAV1-IDS



(**b**) UAV2-IDS



(**c**) UAV3-IDS

Figure 8. Combined performance metrics of CNN-1D models over 25 training epochs for UAV1-IDS, UAV2-IDS, and UAV3-IDS. (**a**) Results for UAV1; (**b**) results for UAV2; (**c**) results for UAV3.

Table 10. Classification results of the CNN-1D model on the UAV2-IDS dataset.

Class	Precision	Recall	F1-Score	Support
Brute-Force	1.000	1.000	1.000	34,502
DDoS	0.570	1.000	0.730	6318
De-authentication	1.000	0.850	0.920	31,098
DoS	1.000	1.000	1.000	4722
Evil Twin	1.000	1.000	1.000	2007
Fake Landing	1.000	1.000	1.000	4917
MITM	1.000	1.000	1.000	1003
Normal	1.000	1.000	1.000	25,908
Reconnaissance	1.000	1.000	1.000	4
Replay	1.000	1.000	1.000	1705
Scanning	1.000	1.000	1.000	14,064
Accuracy			0.960	126,248
Macro Avg	0.960	0.990	0.970	126,248
Weighted Avg	0.980	0.960	0.970	126,248

Table 11. Classification results of the CNN-1D model on the UAV3-IDS dataset.

Class	Precision	Recall	F1-Score	Support
Benign	0.420	0.950	0.580	19
Brute-Force	1.000	0.990	0.990	2032
DDoS	1.000	1.000	1.000	18,773
De-authentication	0.000	0.000	0.000	264
DoS	0.990	1.000	1.000	41,337
Evil Twin	0.000	0.000	0.000	12
Fake Landing	0.470	0.510	0.490	98
MITM	0.810	0.850	0.830	494
Reconnaissance	0.410	1.000	0.580	2857
Scanning	1.000	0.540	0.700	8893
Accuracy			0.940	74,779
Macro Avg	0.610	0.680	0.620	74,779
Weighted Avg	0.970	0.940	0.940	74,779

6.2. Phase 2: Evaluating UAV-IDS-CNN-1D Under Adversarial Attacks Using All Features

This study hypothesizes that while adversarial evasion attacks can successfully craft samples misclassified as normal class, thus degrading the performance of the UAV-IDS model, they often fail to preserve the underlying semantics of the original cyberattack. Consequently, these samples may have no practical impact in a real network environment.

To validate the necessity of DIFS, FGSM and BIM attacks were applied to the three UAV-IDS CNN-1D models using all features without using DIFS. The results demonstrate that, while the attacks achieve substantial ASR, the adversarial samples generated by perturbing all features lack functional or semantic equivalence to realistic cyberattacks. Tables 12–14 summarize adversarial robustness metrics for both initial and retrained models under FGSM and BIM attacks.

The experimental results reveal three key observations. First, the initial models exhibit moderate vulnerability, with ASR values ranging from 4.36% to 13.68% across datasets and attack types. Second, retrained models demonstrate significantly enhanced robustness, reducing ASR to nearly negligible levels (0.03–0.37%) while maintaining a high clean accuracy (91.78–95.82%). In particular, the UAV3-IDS dataset shows the largest improvement, with the BIM Attack Success Rate decreasing from 13.68% to 0.03% after retraining.

Table 12. UAV1-IDS CNN-1D adversarial robustness summary.

Attack	Model	Clean Acc. (%)	Robust Acc. (%)	ASR (%)
BIM	Initial	96.31	95.45	4.55
FGSM	Initial	96.34	95.50	4.50
BIM	Retrained	95.24	99.93	0.07
FGSM	Retrained	95.82	99.71	0.29

Table 13. UAV2-IDS CNN-1D adversarial robustness summary.

Attack	Model	Clean Acc. (%)	Robust Acc. (%)	ASR (%)
FGSM	Initial	95.23	95.64	4.36
BIM	Initial	96.17	94.51	5.49
BIM	Retrained	95.44	99.91	0.09
FGSM	Retrained	93.97	99.63	0.37

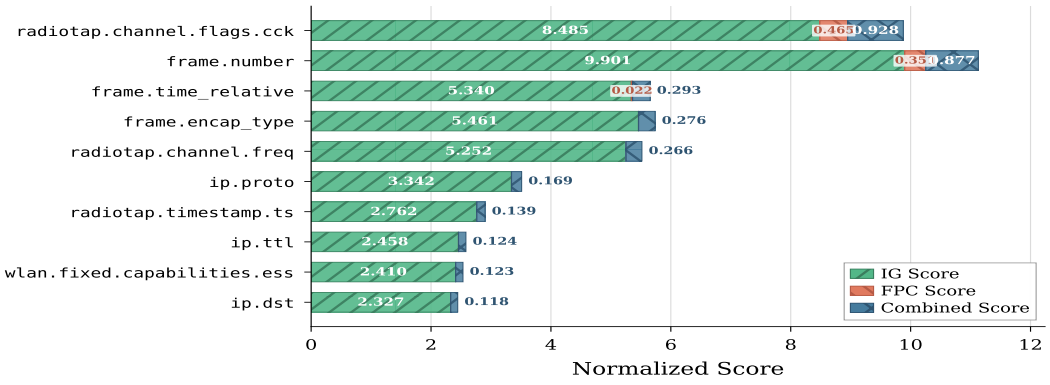
Table 14. UAV3-IDS CNN-1D adversarial robustness summary.

Attack	Model	Clean Acc. (%)	Robust Acc. (%)	ASR (%)
FGSM	Initial	69.54	94.88	5.12
BIM	Initial	78.93	86.32	13.68
BIM	Retrained	93.65	99.97	0.03
FGSM	Retrained	91.78	99.96	0.04

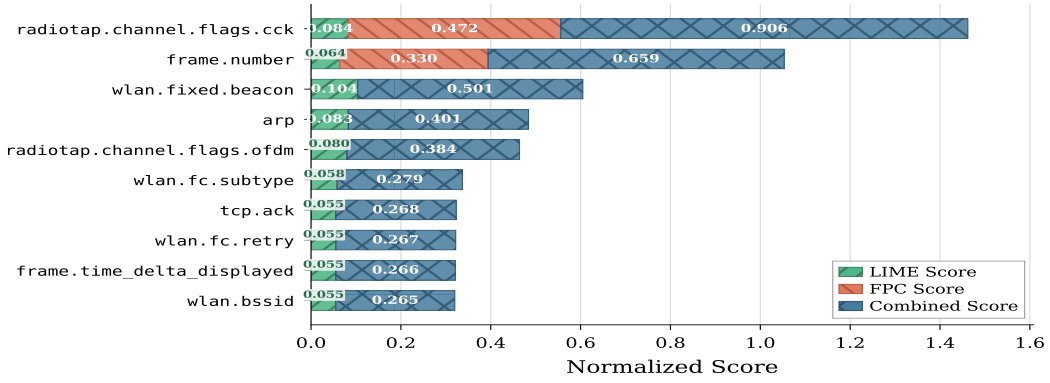
6.3. Phase 3: DIFS Feature Selection Analysis

To optimize adversarial robustness testing, we applied the DIFS method to the three UAV-IDS datasets. DIFS integrates MPI and FPC to identify a refined set of critical features. Figure 9 presents the top 10 features selected by DIFS for the UAV1-IDS dataset, where MPI scores (from IG or LIME) are shown in green, FPC scores in orange, and the final combined DIFS score (with $\alpha = 0.5$) in gray. The results for the UAV2-IDS and UAV3-IDS datasets are provided in Figures 10 and 11, respectively.

This refined feature set was subsequently used to evaluate the robustness of the UAV-IDS CNN-1D model. Adversarial attacks (FGSM and BIM) were configured to perturb only these top 10 selected features.

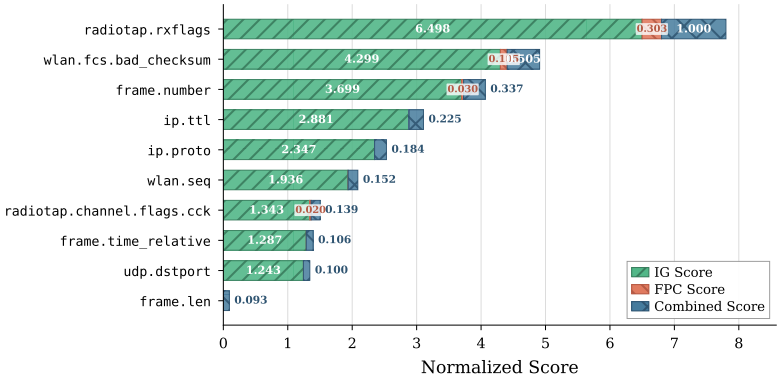


(**a**) Top 10 features using IG and FPC

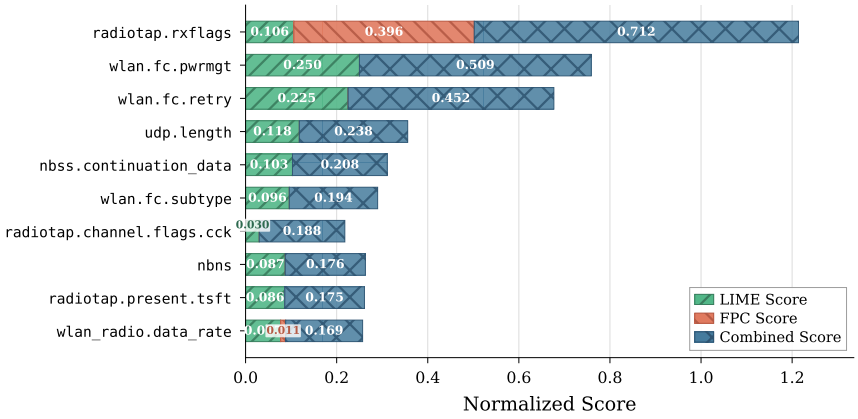


(**b**) Top 10 features using LIME and FPC

Figure 9. Top 10 feature rankings for UAV1-IDS using the DIFS methodology.

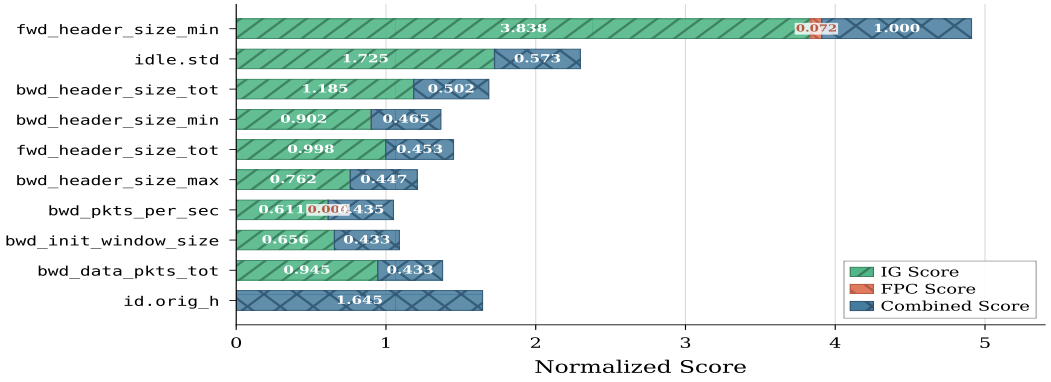


(**a**) Top 10 features using IG and FPC

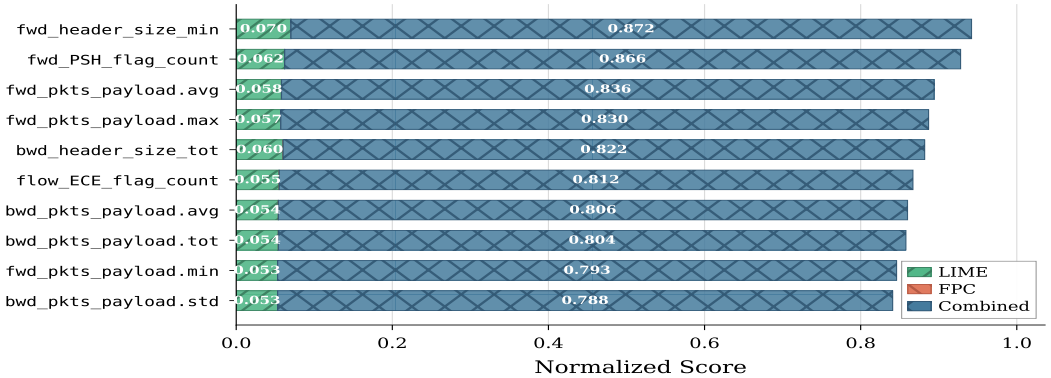


(**b**) Top 10 features using LIME and FPC

Figure 10. Top 10 feature rankings for UAV2-IDS using DIFS methodology.



(**a**) Top 10 features using IG and FPC



(**b**) Top 10 features using LIME and FPC

Figure 11. Top 10 feature rankings for UAV3-IDS using DIFS methodology.

6.4. Phase 4: Evaluating UAV-IDS-CNN-1D Under Adversarial Attacks Using DIFs

This section evaluates the robustness of the proposed UAV-IDS-CNN-1D model against realistic adversarial attacks crafted using the DIFS method. The adversarial strategy perturbs only the top 10 features identified by IG and LIME methods while adhering to FPC, ensuring that generated samples maintain realistic cyberattack patterns that could evade detection in real-world UAV environments.

The experimental methodology comprises two phases. First, initially trained models are evaluated on both clean test samples and adversarial samples crafted using the top 10 DIFS features. This assessment reveals substantial inherent resilience against functionally constrained cyberattacks. Second, models undergo adversarial retraining using augmented datasets that incorporate both clean and adversarial samples to improve generalizability.

The results in Tables 15–17 demonstrate the effectiveness of this approach. Adversarial retraining produces significant improvements in model robustness in all configurations. UAV1-IDS-CNN-1D achieves 99.91% Robust Accuracy against BIM attacks using LIME + FPC features, while UAV2-IDS-CNN-1D achieves 99.93% Robust Accuracy against BIM attacks with IG + FPC features. UAV3-IDS-CNN-1D achieves 99.97% Robust Accuracy against BIM attacks using both feature selection methods.

Table 15. UAV1-IDS-CNN adversarial robustness summary using DIFS.

Attack	Feature Method	Model	Clean Accuracy	Robust Accuracy/ASR
FGSM	IG + FPC	Initial	0.9626	0.9691/0.0309
		Retrained	0.9105	0.9936/0.0064
	LIME + FPC	Initial	0.9630	0.9575/0.0425
		Retrained	0.9580	0.9991/0.0009
BIM	IG + FPC	Initial	0.9649	0.9649/0.0351
		Retrained	0.9565	0.9993/0.0007
	LIME + FPC	Initial	0.9649	0.9532/0.0468
		Retrained	0.9619	0.9999/0.0001

Table 16. UAV2-IDS-CNN adversarial robustness summary using DIFS.

Attack	Feature Method	Model	Clean Accuracy	Robust Accuracy/ASR
FGSM	IG + FPC	Initial	0.9527	0.9875/0.0125
		Retrained	0.7069	0.9314/0.0686
	LIME + FPC	Initial	0.9595	0.9647/0.0353
		Retrained	0.9666	0.9983/0.0017
BIM	IG + FPC	Initial	0.9602	0.9746/0.0254
		Retrained	0.9507	0.9993/0.0007
	LIME + FPC	Initial	0.9591	0.9847/0.0153
		Retrained	0.9508	0.9966/0.0034

ASR drops to minimal levels across all models, with several configurations achieving ASR values below 0.1%. These results validate the robustness of the proposed approach, successfully hardening UAV-IDS-CNN-1D against sophisticated evasion techniques while maintaining a reliable performance on legitimate network traffic. However, some configurations exhibit notable trade-offs that warrant discussion. The UAV2-IDS model with IG + FPC under FGSM shows a clean accuracy reduction from 95.27% to 70.69% after retraining, indicating that IG-based feature selection may produce overly conservative decision boundaries for certain dataset–attack combinations. Similarly, the UAV3-IDS models retain a lower clean accuracy (73.54–82.29%) compared to UAV1-IDS and UAV2-IDS, reflecting the inherent difficulty of the GCS-based attack scenario with its larger feature space

(85 features). These results indicate that robustness improvements are not uniformly cost-free, and practitioners should evaluate the trade-off between clean accuracy and robustness for their specific deployment context.

Table 17. UAV3-IDS-CNN adversarial robustness summary using DIFS.

Attack	Feature Method	Model	Clean Accuracy	Robust Accuracy/ASR
FGSM	IG + FPC	Initial	0.6687	0.9976/0.0024
		Retrained	0.9390	0.9991/0.0009
	LIME + FPC	Initial	0.6482	0.9780/0.0220
		Retrained	0.7354	0.9944/0.0056
BIM	IG + FPC	Initial	0.7926	0.9557/0.0443
		Retrained	0.7752	0.9997/0.0003
	LIME + FPC	Initial	0.7711	0.9646/0.0354
		Retrained	0.8229	0.9997/0.0003

7. Discussion

7.1. Interpretation of Key Findings

The experimental results demonstrate several significant aspects of adversarial robustness in UAV-IDS CNN-1D models. The remarkable reduction in ASR from initial levels of 4.36–13.68% to nearly negligible values (0.02–0.37%) after adversarial retraining underscores the effectiveness of the proposed DIFS methodology. This substantial improvement can be attributed to three primary factors.

Firstly, strategic feature selection through DIFS enabled focusing perturbation on the most critical features, creating more meaningful adversarial examples during retraining. The consistent performance across the three UAV datasets (UAV1-IDS, UAV2-IDS, and UAV3-IDS) validates the generalizability of this approach.

Secondly, the maintained clean accuracy (91.78–95.82%) alongside improved Robust Accuracy (99.63–99.97%) indicates that adversarial retraining does not compromise the model's performance on legitimate traffic while significantly enhancing its defensive capabilities. Table 18 illustrates the final checklist for all UAV-IDS-CNN-1D models.

Table 18. Performance checklist for UAV-IDS-CNN models.

Model Configuration	Clean Accuracy	Robust Accuracy	Low ASR
UAV1 FGSM LIME + FPC	✓	✓	✓
UAV1 BIM LIME + FPC	✓	✓	✓
UAV2 FGSM LIME + FPC	✓	✓	✓
UAV2 BIM IG + FPC	✓	✓	✓
UAV3 FGSM IG + FPC	✓	✓	✓
UAV3 BIM IG + FPC	×	✓	✓
UAV3 BIM LIME + FPC	×	✓	✓

7.2. Expected Trade-Offs in the DIFS Framework

The DIFS framework involves deliberate trade-offs designed to optimize adversarial robustness and operational feasibility. Feature Space Reduction vs. Attack Coverage: By selecting only the top-10 DIFs (22% of features in UAV1-IDS to 12% in UAV3-IDS), we intentionally reduce the attack surface for adversarial perturbation. This trade-off is strategically desirable for three reasons: Firstly, it forces adversarial training to focus on realistic, functionality-preserving attacks rather than protocol-violating perturbations. Secondly, focusing on 10 features reduces computational complexity, making attacks tractable for

both red teams and defenders [24]. Thirdly, security practitioners can more easily monitor the top-10 critical features in deployed systems.

Clean Accuracy vs. Robust Accuracy: Adversarial retraining on DIFS-constrained samples may cause slight reductions in clean accuracy as the model learns more conservative decision boundaries [27]. This is most evident in the UAV3-IDS dataset, where clean accuracy decreased from 78.93% to 77.52% in some configurations. However, this trade-off is not only acceptable but desirable in high-security UAV deployments. The corresponding improvement in Robust Accuracy (86.32% → 99.97%) and ASR reduction (13.68% → 0.03%) far outweighs the minor clean accuracy loss. In adversarial environments, clean accuracy alone is misleading—an IDS with 95% clean accuracy but 50% ASR provides little practical security.

Computational Efficiency vs. Comprehensiveness: Focusing adversarial training on 10 DIFs rather than all features (45–85) yields substantial computational benefits: the training time is approximately 4–8× faster, the inference overhead is significantly reduced and the memory footprint is 78–88% smaller [5]. This trade-off accepts that adversaries might find evasion opportunities in non-DIF features, but such evasions would necessarily be functionality-violating.

Table 19 summarizes the expected and observed performance changes.

Table 19. Expected vs. observed DIFS performance outcomes.

Metric	Expected	Observed	Evaluation
Robust Accuracy	Increase	99%+	✓ Achieved
ASR Reduction	Decrease	Near-zero	✓ Achieved
Efficiency	Improve	4–8× speedup	✓ Achieved
Clean Accuracy	Slight decrease	0–2% drop	✓ Acceptable
Interpretability	Improve	10 vs. 45–85	✓ Achieved

The DIFS framework optimizes the balance between adversarial robustness and operational feasibility—a trade-off that strongly favors deployment in real-world UAV security scenarios.

7.3. Implications of Testbed-Generated Data for Model Generalizability

The use of physically generated UAV-NIDD datasets [29] has significant implications for model generalizability. Unlike simulated datasets, testbed data captures authentic protocol behaviors, physical layer properties, and genuine attack execution patterns [5]. This realism provides three key advantages:

1. Protocol Fidelity: Network features reflect actual physical layer characteristics, MAC-layer timing, and real hardware constraints. For example, RSSI values, packet inter-arrival times, and retransmission patterns in the UAV-NIDD datasets represent genuine wireless channel behavior rather than idealized simulations [46].
2. Attack Realism: Adversarial samples represent executable attacks that succeeded against actual UAV hardware and network stacks. This ensures our robust models defend against threats that can genuinely compromise deployed systems, not just theoretical attack vectors.
3. Hardware Variability: Natural variations in testbed equipment (different radio chipsets, antenna configurations, and environmental RF conditions) introduce diversity that aids cross-platform generalization [4].

However, generalization boundaries exist. Models trained on DJI Mavic Air 2 traffic patterns may require adaptation for significantly different UAV architectures, though

commercial platforms sharing similar protocol stacks (MAVLink, DJI SDK) should exhibit strong cross-platform generalization. Attack coverage is limited to the cyberattack types present in UAV-NIDD, including DDoS, DoS, UDP Flooding, ICMP Flooding, Deauthentication, Brute-Force, Evil Twin, Scanning, Reconnaissance, GPS Jamming, Fake Landing, MITM, and Replay; emerging attack classes would necessitate dataset updates and model retraining.

Environmental factors present additional challenges. The UAV-NIDD testbed [29] operated in controlled indoor/outdoor conditions with known RF interference profiles. Deployment in different electromagnetic environments (urban canyons, military jamming scenarios, and extreme weather) may require online adaptation mechanisms or periodic retraining with location-specific data [30].

Estimated Generalization Boundaries: Based on the UAV-NIDD dataset characteristics and our model architecture, we estimate the following generalization boundaries. These estimates are based on architectural similarity and protocol overlap, and have not been empirically validated on external datasets:

- Strong generalization (>95% accuracy): Commercial UAVs using MAVLink or DJI SDK protocols with similar frequency bands (2.4 GHz, 5.8 GHz). Examples: DJI Phantom series, Parrot Anafi, and 3DR Solo.
- Good generalization (85–95% accuracy): UAV swarms and multi-platform networks with transfer learning or fine-tuning on small platform-specific validation sets (100–500 samples). The DIFS feature selection approach remains valid even if specific feature distributions shift.
- Moderate generalization (70–85% accuracy): Custom military or industrial UAV systems with proprietary protocols. These would require fine-tuning on domain-specific data, though the dual-criteria feature selection methodology transfers directly.
- Limited generalization (<70% accuracy): Non-UAV IoT networks or ground vehicle telematics. While the DIFS methodology (combining MPI and FPC) remains applicable as a framework, the specific DIFs identified would not transfer. New feature selection would be required.

The achieved performance metrics in this work (99.8% clean accuracy, ASR reduction to 0.002) reflect both the quality of testbed-generated training data and the effectiveness of functionality-preserving adversarial training. We emphasize that these results reflect performance on UAV network traffic that adheres to the protocol and that reflects the attack characteristics present in the UAV-NIDD datasets. Practitioners deploying our approach in significantly different UAV environments should budget for validation testing and potential fine-tuning, though the core DIFS framework provides a solid foundation for adversarial robustness across diverse platforms.

7.4. Comparative Analysis with the Existing Literature

Compared to conventional adversarial training approaches that alter all features [10,14,32,47], our DIFS method demonstrates improved efficiency. The focused perturbation on the top-10 features achieved comparable or better robustness while reducing computational overhead and maintaining the semantic validity of adversarial samples.

The Robust Accuracy (99–99.97%) against BIM attacks compares favorably with reported results in similar Intrusion Detection Systems [29], particularly noting the challenging nature of UAV network environments with their unique traffic patterns and constraints.

7.5. Methodological Contributions

This work contributes to the practical and methodological understanding of adversarial robustness in the following ways. The DIFS methodology provides a framework for

balancing feature importance with functional constraints, addressing the semantic validity challenge in adversarial sample generation. The consistent performance improvement across datasets with different feature counts (45, 51, and 85) demonstrates the approach's adaptability within the evaluated UAV-IDS domain. Furthermore, the success of targeted feature perturbation suggests that strategic, focused perturbations can yield results that are competitive or superior to those achieved with comprehensive feature coverage, at least within the scope of the datasets and attack models examined here.

8. Conclusions

This paper presented a four-phase framework for evaluating and improving the adversarial robustness of UAV-IDS-CNN-1D models. The methodology integrates DIFS with adversarial training, demonstrating a significant improvement in model resilience against evasion attacks. Three distinct UAV-IDS-CNN-1D models were constructed and evaluated across three UAV-IDS datasets that represent different cyberattack scenarios targeting the network components of a real UAV testbed. Multiple DIFS configurations (combining IG or LIME with FPC) were used to identify the most critical functional features. These features were then used to execute two constrained adversarial attacks, FGSM and BIM, generating functional cyber–semantic samples. These samples served a dual purpose: first, to evaluate the initial UAV-IDS-CNN-1D models, and second, to conduct adversarial retraining as the final robustness phase. The experimental results indicate that the initial UAV-IDS-CNN-1D models already exhibit considerable robustness, with only a slight drop in Robust Accuracy and ASR. Furthermore, adversarial retraining using only DIF-selected samples proved highly effective, driving the ASR to near-zero levels and achieving a Robust Accuracy of 99–99.98% across all models. These findings suggest that an approach focused on the top functional features can yield robust UAV-IDS-CNN-1D models, though further validation across multiple UAV platforms and attack scenarios is needed to establish broader generalizability.

Future research will explore several promising directions. Firstly, we plan to extend the proposed framework to handle streaming data and operate in online learning scenarios for real-time intrusion detection. Secondly, we will investigate more complex black-box attack strategies, leveraging generative models and metaheuristic approaches such as Genetic Algorithms (GAs) and PSO at an abstract feature level. Finally, a deeper analysis of the functionality and semantic preservation of adversarial samples for individual cyberattack classes is warranted to develop more nuanced and effective defense mechanisms.

Author Contributions: Conceptualization, S.A. and M.A.-F.; methodology, S.A. and M.A.-F.; software, S.A.; validation, S.A., M.A.-F. and M.A.; formal analysis, S.A.; investigation, S.A.; resources, M.A.-F. and M.A.; data curation, S.A.; writing—original draft preparation, S.A.; writing—review and editing, M.A.-F. and M.A.; visualization, S.A.; supervision, M.A.-F. and M.A.; project administration, M.A.-F. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data and source code supporting the findings of this study are openly available in the DIFS-UAV-IDS repository accessed on 10 May 2026. https://github.com/saleemsar/UAV-ADV/blob/main/UAV_Functional.ipynb. This includes the implementation of the DIFS framework, the CNN-1D model architectures, and preprocessed versions of the UAV network intrusion datasets used for evaluation.

DURC Statement: Current research is limited to the field of cybersecurity and network intrusion detection for unmanned aerial vehicles (UAVs), which is beneficial for protecting critical UAV infrastructure, ensuring safe autonomous operations, and safeguarding civilian and military communication networks against cyber threats, and does not pose a threat to public health or national

security. Authors acknowledge the dual-use potential of the research involving adversarial attack patterns and evasion techniques against IDS frameworks and confirm that all necessary precautions have been taken to prevent potential misuse. As an ethical responsibility, authors strictly adhere to relevant national and international laws about DURC. Authors advocate for responsible deployment, ethical considerations, regulatory compliance, and transparent reporting to mitigate misuse risks and foster beneficial outcomes.

Acknowledgments: The authors sincerely acknowledge the Princess Sumaya University for Technology for supporting the steps of this work.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

Abbreviations used in the DIFS framework and UAV security context. The following abbreviations are used in this manuscript:

Abbreviation	Full Definition
ASR	Attack Success Rate
BIM	Basic Iterative Method
C-BIM	Constrained Basic Iterative Method
C-FGSM	Constrained Fast Gradient Sign Method
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DIFS	Dual-Impact Feature Selection
DIFs	Dual-Impact Features
DL	Deep Learning
DoS	Denial of Service
FGSM	Fast Gradient Sign Method
FPC	Functionality Preservation Criterion
GCS	Ground Control Station
IDS	Intrusion Detection System
IG	Integrated Gradients
IoT	Internet of Things
LIME	Local Interpretable Model-agnostic Explanations
MITM	Man-in-the-Middle
ML	Machine Learning
MPI	Model Performance Impact
PGD	Projected Gradient Descent
UAV	Unmanned Aerial Vehicle
XAI	Explainable AI

References

1. Hadi, H.J.; Cao, Y.; Nisa, K.U.; Jamil, A.M.; Ni, Q. A comprehensive survey on security, privacy issues and emerging defence technologies for UAVs. *J. Netw. Comput. Appl.* **2023**, *213*, 103607. [CrossRef]
2. Zhao, L.; Li, S.; Tan, Z.; Hawbani, A.; Timotheou, S.; Yu, K. A Multi-UAV Cooperative Task Scheduling in Dynamic Environments: Throughput Maximization. *IEEE Trans. Comput.* **2025**, *74*, 442–454. [CrossRef]
3. Zhou, Z.; Tang, J.; Feng, W.; Zhao, N.; Yang, Z.; Wong, K.K. Optimized routing protocol through exploitation of trajectory knowledge for uav swarms. *IEEE Trans. Veh. Technol.* **2024**, *73*, 15499–15512. [CrossRef]
4. Lu, X.; Xiao, L.; Dai, C.; Dai, H. UAV-aided cellular communications with deep reinforcement learning against jamming. *IEEE Wirel. Commun.* **2020**, *27*, 48–53. [CrossRef]
5. Hadi, H.J.; Cao, Y.; Li, S.; Xu, L.; Hu, Y.; Li, M. Real-time fusion multi-tier DNN-based collaborative IDPS with complementary features for secure UAV-enabled 6G networks. *Expert Syst. Appl.* **2024**, *252*, 124215.
6. Braun, G.; Cohen, S.; Rokach, L. Adversarial evasion attacks detection for tree-based ensembles: A representation learning approach. *Inf. Fusion* **2025**, *118*, 102964. [CrossRef]

7. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 2154–2156.
8. Li, Z.; Qiu, Y. Feature selection based on improved principal component analysis. In Proceedings of the 2023 2nd Asia Conference on Algorithms, Computing and Machine Learning, Shanghai, China, 17–19 March 2023; pp. 188–192.
9. Bernard, S.; Heutte, L.; Adam, S. On the selection of decision trees in random forests. In Proceedings of the 2009 International Joint Conference on Neural Networks, IEEE, Atlanta, GA, USA, 14–19 June 2009; pp. 302–307.
10. Dhal, P.; Azad, C. A comprehensive survey on feature selection in the various fields of machine learning. *Appl. Intell.* **2022**, *52*, 4543–4581.
11. Zhang, L.; Bai, Y.; Xue, T.; Feng, G.; Zhang, H. EfficientSTNet: A Deep Learning Approach for Multi-Class DDoS Detection. *IEEE Access* **2025**, *13*, 157587–157599. [CrossRef]
12. Ribeiro, M.T.; Singh, S.; Guestrin, C. "Why should i trust you?" Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
13. Park, C.; Lee, J.; Kim, Y.; Park, J.G.; Kim, H.; Hong, D. An Enhanced AI-Based Network Intrusion Detection System Using Generative Adversarial Networks. *IEEE Internet Things J.* **2023**, *10*, 2330–2345. [CrossRef]
14. Akbar, M.R.; Nuha, H.H.; Mugitama, S.A. Intrusion Detection on Unmanned Aerial Vehicle (UAV) using Binary Decision Tree. In Proceedings of the 2023 11th International Conference on Information and Communication Technology (ICoICT), Melaka, Malaysia, 23–24 August 2023; pp. 633–638. [CrossRef]
15. Srinivasan, M.; Senthilkumar, N.C. Intrusion Detection and Prevention System (IDPS) Model for IIoT Environments Using Hybridized Framework. *IEEE Access* **2025**, *13*, 26608–26621. [CrossRef]
16. Whelan, J.; Sangarapillai, T.; Minawi, O.; Almehmadi, A.; El-Khatib, K. Novelty-based Intrusion Detection of Sensor Attacks on Unmanned Aerial Vehicles. In Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, Q2SWinet '20, New York, NY, USA, 16–20 November 2020; pp. 23–28. [CrossRef]
17. Gao, Y.; Liu, Y.; Jin, Y.; Chen, J.; Wu, H. A novel semi-supervised learning approach for network intrusion detection on cloud-based robotic system. *IEEE Access* **2018**, *6*, 50927–50938.
18. Li, Y.; Lin, J.; Xiong, K. An adversarial attack defending system for securing in-vehicle networks. In Proceedings of the 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), Las Vegas, NV, USA, 9–12 January 2021; pp. 1–6.
19. Aryal, K.; Gupta, M.; Abdelsalam, M.; Saleh, M. Explainability Guided Adversarial Evasion Attacks on Malware Detectors. In Proceedings of the 2024 33rd International Conference on Computer Communications and Networks (ICCCN), Kailua-Kona, HI, USA, 29–31 July 2024; pp. 1–9. [CrossRef]
20. Kwon, H.; Yoon, H.; Park, K.W. Robust captcha image generation enhanced with adversarial example methods. *IEICE Trans. Inf. Syst.* **2020**, *103*, 879–882. [CrossRef]
21. Wang, B.; Zou, F.; Liu, X. New algorithm to generate the adversarial example of image. *Optik* **2020**, *207*, 164477. [CrossRef]
22. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations (ICLR 2014), Banff, AB, Canada, 14–16 April 2014.
23. Tian, Z.; Cui, L.; Liang, J.; Yu, S. A comprehensive survey on poisoning attacks and countermeasures in machine learning. *ACM Comput. Surv.* **2022**, *55*, 1–35. [CrossRef]
24. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
25. Aryal, K.; Gupta, M.; Abdelsalam, M. Analysis of label-flip poisoning attack on machine learning based malware detector. In Proceedings of the 2022 IEEE International Conference on Big Data (Big Data), Osaka, Japan, 17–20 December 2022; pp. 4236–4245.
26. Kurakin, A.; Goodfellow, I.J.; Bengio, S. Adversarial examples in the physical world. In *Artificial Intelligence Safety and Security*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2018; pp. 99–112.
27. Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; Madry, A. Adversarial examples are not bugs, they are features. *Adv. Neural Inf. Process. Syst.* **2019**, *32*, 8307.
28. Sauka, K.; Shin, G.Y.; Kim, D.W.; Han, M.M. Adversarial robust and explainable network intrusion detection systems based on deep learning. *Appl. Sci.* **2022**, *12*, 6451. [CrossRef]
29. Hadi, H.J.; Cao, Y.; Khan, M.K.; Ahmad, N.; Hu, Y.; Fu, C. UAV-NIDD: A Dynamic Dataset for Cybersecurity and Intrusion Detection in UAV Networks. *IEEE Trans. Netw. Sci. Eng.* **2025**, *12*, 2739–2757. [CrossRef]
30. Ferrag, M.A.; Maglaras, L.; Moschoyiannis, S.; Janicke, H. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. *J. Inf. Secur. Appl.* **2020**, *50*, 102419. [CrossRef]
31. Basnet, R.B.; Shash, R.; Johnson, C.; Walgren, L.; Doleck, T. Towards Detecting and Classifying Network Intrusion Traffic Using Deep Learning Frameworks. *J. Internet Serv. Inf. Secur.* **2019**, *9*, 1–17.
32. Nag, K.; Pal, N.R. Genetic programming for classification and feature selection. In *Evolutionary and Swarm Intelligence Algorithms*; Springer: Berlin/Heidelberg, Germany, 2018; pp. 119–141.

33. Almomani, O. A feature selection model for network intrusion detection system based on PSO, GWO, FFA and GA algorithms. *Symmetry* **2020**, *12*, 1046. [CrossRef]
34. Han, Y.; Wang, Y.; Cao, Y.; Geng, Z.; Zhu, Q. A Novel Wrapped Feature Selection Framework for Developing Power System Intrusion Detection Based on Machine Learning Methods. *IEEE Trans. Syst. Man Cybern. Syst.* **2023**, *53*, 7066–7076. [CrossRef]
35. Wu, Y.; Yang, L.; Zhang, L.; Nie, L.; Zheng, L. Intrusion Detection for Unmanned Aerial Vehicles Security: A Tiny Machine Learning Model. *IEEE Internet Things J.* **2024**, *11*, 20970–20982. [CrossRef]
36. Chen, X.; Liu, M.; Wang, Z.; Wang, Y. Explainable deep learning-based feature selection and intrusion detection method on the internet of things. *Sensors* **2024**, *24*, 5223. [CrossRef] [PubMed]
37. Niu, A.; Zhang, K.; Zhang, C.; Zhang, C.; Kweon, I.S.; Yoo, C.D.; Zhang, Y. Fast adversarial training with noise augmentation: A unified perspective on randstart and gradalign. *arXiv* **2022**, arXiv:2202.05488.
38. Yang, K.; Liu, J.; Zhang, C.; Fang, Y. Adversarial examples against the deep learning based network intrusion detection systems. In Proceedings of the MILCOM 2018—2018 IEEE Military Communications Conference (MILCOM), IEEE, Los Angeles, CA, USA, 29–31 October 2018; pp. 559–564.
39. Wang, N.; Chen, Y.; Xiao, Y.; Hu, Y.; Lou, W.; Hou, Y.T. Manda: On adversarial example detection for network intrusion detection system. *IEEE Trans. Dependable Secur. Comput.* **2022**, *20*, 1139–1153. [CrossRef]
40. Peng, Y.; Su, J.; Shi, X.; Zhao, B. Evaluating deep learning based network intrusion detection system in adversarial environment. In Proceedings of the 2019 IEEE 9th International Conference on Electronics Information and Emergency Communication (ICEIEC), Beijing, China, 12–14 July 2019; pp. 61–66.
41. Debicha, I.; Debatty, T.; Dricot, J.M.; Mees, W. Adversarial training for deep learning-based intrusion detection systems. *arXiv* **2021**, arXiv:2104.09852.
42. Papadopoulos, P.; Thornewill von Essen, O.; Pitropakis, N.; Chrysoulas, C.; Mylonas, A.; Buchanan, W.J. Launching adversarial attacks against network intrusion detection systems for iot. *J. Cybersecur. Priv.* **2021**, *1*, 252–273. [CrossRef]
43. Kiranyaz, S.; Avci, O.; Abdeljaber, O.; Ince, T.; Gabbouj, M.; Inman, D.J. 1D convolutional neural networks and applications: A survey. *Mech. Syst. Signal Process.* **2021**, *151*, 107398. [CrossRef]
44. Abdeljaber, O.; Avci, O.; Kiranyaz, S.; Gabbouj, M.; Inman, D.J. Real-time vibration-based structural damage detection using one-dimensional convolutional neural networks. *J. Sound Vib.* **2017**, *388*, 154–170. [CrossRef]
45. Vinayakumar, R.; Alazab, M.; Soman, K.P.; Poornachandran, P.; Al-Nemrat, A.; Venkatraman, S. Deep learning approach for intelligent intrusion detection system. *IEEE Access* **2019**, *7*, 41525–41550. [CrossRef]
46. Jalil Hadi, H.; Cao, Y.; Li, S.; Hu, Y.; Wang, J.; Wang, S. Real-Time Collaborative Intrusion Detection System in UAV Networks Using Deep Learning. *IEEE Internet Things J.* **2024**, *11*, 33371–33391. [CrossRef]
47. Lin, Z.; Shi, Y.; Xue, Z. Idsgan: Generative adversarial networks for attack generation against intrusion detection. In *Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 79–91.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.