2025 International Conference on Intelligent Computing
July 26-29, Ningbo, China
https://www.ic-icc.cn/2025/index.php

A Lightweight Real-time Detection Algorithm for Drone WiFi Hijacking

Jingxian Zhou[1] and Chengcheng Shangguan[2][0009-0003-9678-8452]

[1] Information Security Evaluation Center, Civil Aviation University of China, 300300 Tianjin, China.
[2]College of Safety Science and Engineering, Civil Aviation University of China, 300300 Tianjin, China
sgcc0619@163.com

Abstract. Due to WiFi protocols prioritized usability over security and adopted weak encryption because of resource constraints，consumer drones are vulnerable to malicious hijacking attacks when communicating with ground stations via WiFi. In order to carry out drone WiFi hijacking attack experiments, the research team built the first real drone WiFi hijacking attack dataset to address the scarcity of real attack samples in this field, covering multiple types such as De-Authentication attacks. At the same time, considering the limited computing resources of drones and the high real-time requirements of communications, the team used self-built datasets and public datasets to conduct multi-dimensional comparative experiments on existing algorithms, selected the XGBoost model that takes into account both detection accuracy and lightness as the basic framework, designed a three-level feature screening mechanism of "variance threshold filtering-high correlation elimination-Boruta feature selection", and introduced a weighted cross entropy loss function to optimize learning performance, and developed a lightweight drone WiFi hijacking real-time detection algorithm. The experimental results show that this method can effectively detect drone WiFi hijacking attack traffic, and its comprehensive performance is better than the existing algorithms; compared with the original XGBoost method, the accuracy of the proposed method reaches 96.3%, and the inference time is shortened by half, which has both high accuracy and lightness.

Keywords: UAV WiFi Hijacking, Intrusion Detection System, Feature Selection Mechanism, Weighted Cross-Entropy.

1 Introduction

In recent years, the low-altitude [1] economy industry has garnered policy support and has been actively implemented across regions, entering a phase of rapid development. As a pivotal component of the low-altitude economy [2], unmanned aerial vehicles (UAVs) have witnessed significant growth in recent times. China boasts the largest consumer UAV market globally, with a market size estimated at approximately 327.5

billion yuan in 2024. This is projected to surge to nearly 50 billion yuan in 2025, marking an annual growth rate exceeding 50%. Amidst the rapid expansion of the UAV industry, the issue of low-altitude safety has garnered increasing attention, leading to a burgeoning anti-UAV market and a rising demand for the establishment of UAV WiFi security regulatory frameworks.

In safety-critical UAV operations, communication architectures are categorized into three key types [3]. Satellite-based systems enable precise positioning via global networks, offering high reliability and interference resistance in remote/complex environments despite high deployment/maintenance costs. UAV self-organizing Mesh networks support direct node-to-node communication with dynamic routing capabilities, ensuring mission continuity through automatic path reconfiguration when failures occur. Short-range wireless links (e.g., Bluetooth/WiFi) facilitate flexible near-field data exchange with ground stations, though they suffer from signal interference and unstable transmission.

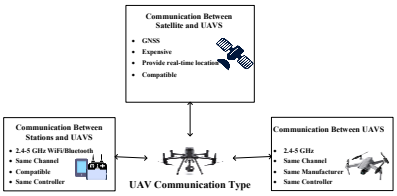

Fig. 1. Primary Communication Types of Unmanned Aerial Vehicles (UAVs)

As a standard drone communication tech, WiFi creates a bidirectional air-ground data link via high-frequency signals. Core functions: 1) high-precision flight control—using a dedicated remote to manage trajectories/camera angles and interact real-time with ground stations; 2) multi-mode data transmission—real-time low-bandwidth flight/sensor data alongside 60Mbps 4K/8K video over 5GHz, maintaining 5–10km line-of-sight range and supporting mobile apps for flight control and multi-camera coordination.

Although drones leverage common WLAN protocols for flexible communication, they also inherit the intrinsic security vulnerabilities of open wireless transmission [4]. The broadcast nature of wireless channels makes their communication links susceptible to threats including man-in-the-middle attacks, DoS flooding, DE authentication frame injection, and MAC address spoofing. Attackers may intercept control commands or navigation data, or even forge signals to hijack drones. Compared with traditional WLAN applications, the dynamic flight characteristics of drones further expand the attack surface, significantly escalating risks of illegal access and data tampering in airspace environments.

Drone networks are severely constrained by limited device resources, rendering traditional encryption methods impractical due to their high computational costs. Although emerging technologies like blockchain, software-defined networks (SDNs), machine learning, and fog/edge computing have been integrated into architectural frameworks, resource-constrained drones still cannot run complex security algorithms. Given that gateways often function as fog nodes, developing lightweight host-based intrusion

detection systems (HIDS) tailored for low-computational environments becomes crucial. These systems aim to minimize resource usage while enabling real-time threat monitoring, ensuring optimal security performance in drone networks.

2 Related works

The UAV network is confronted with numerous security threats and issues. In recent years, intrusion detection systems (IDS) have been deployed to detect malicious activities of drones and identify potential suspicious attacks targeting them. Typically, IDS monitors both incoming and outgoing network traffic, analyzing it to detect anomalies. Their objective is to detect and recognize cyber-attacks by scrutinizing data audits gathered from various segments of the network. Moving forward, several IDS methods aimed at safeguarding the UAV network from intruders will be introduced.

Significant advancements in rule-based intrusion detection systems (RB-IDS) for drones have been done: Strohmeier et al. [7] developed a false data injection attack detection scheme leveraging signal strength analysis in drone-ground station communications, achieving attack identification within 40 seconds. However, critical limitations persist: dynamic rule base maintenance relies heavily on manual updates, increasing operational complexity, and detection mechanisms dependent on static known-attack signature libraries inherently fail to address emerging unknown threats, creating persistent detection gaps.

Anomaly detection-based intrusion detection systems (IDS) are also applied in drone security to counter interference attacks. Condomines et al. [8] proposed a hybrid IDS architecture integrating spectrum traffic analysis with FANET robust control algorithms for real-time distributed DoS attack estimation. Their experiments demonstrated accurate identification of multiple anomaly types and significant detection performance for distributed DoS attacks. However, current research still requires multi-scenario long-term validation to confirm reliability in practical deployments.

Existing real-time intrusion detection system (IDS) studies exhibit notable limitations. Zang and Yan [9] achieved a 0.61-second detection time using a random forest-based IDS for vehicular ad-hoc networks on Mininet-WiFi, yet their approach lacked optimization for drone scenarios. Das et al. [10] developed a Bagging Extra Trees/RF/XGBoost ensemble model achieving 98.26% accuracy on the AWID dataset (covering injection, flooding, spoofing attacks, and normal traffic), but its 13.03-second training time severely compromised real-time performance. Ahmad et al. [11] attained a 432ms detection speed via a low-latency feature selection mechanism using random forest-sorted features on NSL-KDD, though their model failed to adapt to WiFi traffic. Collectively, these methods inadequately address both real-time requirements and environmental adaptability challenges specific to drone operation scenarios.

Literature [12] proposed an LSTM-based algorithm achieving high accuracy on CICIDS2017, NSL-KDD, and UNSW-NB15, but it didn't quantify training time or detection speed. Reference [13] introduced FEDGAN-IDS, a privacy-preserving IDS combining GAN and federated learning, achieving 98.7% attack detection accuracy. However, its datasets (NSL-KDD, KDD-CUP99, UNSW-NB15) lack radio/MAC layer

features, making it unsuitable for direct WiFi attack classification. Research on IEEE 802.11 networks [14-15] shows that although deep learning methods can significantly improve classification accuracy, they have significant defects: the complex model architecture makes the training/testing/inference process time-consuming, and hyperparameter tuning will cause optimization delays. In contrast, the lightweight IDS architecture based on XGBoost has more advantages. By simplifying feature engineering and model structure, it achieves higher computing efficiency while ensuring detection effectiveness.

The literature review highlights that existing lightweight anomaly detection models demonstrate insufficient effectiveness in real-time WiFi networks, primarily due to their time complexity ranging from tens of seconds to minutes, which fails to meet the demands of high-speed traffic analysis. A significant research gap persists in real-time detection technologies for high-speed WiFi traffic within highly constrained computing environments, particularly in edge nodes like drone networks. There is an urgent need for lightweight host anomaly detection techniques to minimize computational resource consumption, which can be achieved through optimized machine learning frameworks or low-dimensional statistical approaches.

In this paper, we propose a machine learning detection algorithm for drone WiFi hijacking based on feature selection and loss function optimization. Our contributions mainly include the following four points:

1) We constructed the first real-world dataset of drone WiFi hijacking attacks, including attack types such as MITM attacks and DE authentication attacks, filling the gap in the field of lack of real attack samples on drone WiFi.

2) We conducted multi-method comparison experiments and verified common machine learning algorithms and deep learning algorithms on self-built and public datasets. We found that XGBoost achieved the best balance between detection accuracy and model lightweight, and finally selected this model as the base model.

3) We designed a three-level feature screening mechanism (variance threshold filtering → high correlation elimination → Boruta feature selection) to obtain the globally optimal features of the association, and innovatively introduced the weighted cross entropy loss function to effectively enhance the learning performance.

4) The detection system we built realizes real-time detection of multiple types of attacks and maintains an accuracy rate of 96.3% in mixed attack scenarios, meeting the low latency and high reliability requirements of drone intensive operation scenarios.

3 Materials and methods

In this section, we will describe the experimental dataset required for this study and the strategy for algorithm improvement.

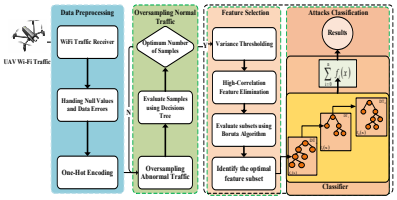

Fig. 2. The detection method diagram of UAV WiFi hijacking.

3.1 Data preprocessing

Data preprocessing is the cornerstone of the data integration pipeline, covering data consolidation, cleaning, transformation, and normalization. In real - world operations, WiFi - enabled UAVs face complex signal interference, causing the collected datasets to have many missing values and noise. These issues make the raw data unfit for machine - learning models and put a heavy computational load on resource - constrained UAVs. To solve this, we use pandas' functions to methodically remove missing values during the cleaning process. Moreover, the collected UAV WiFi traffic data includes unstructured categorical features (such as packet payloads, frame lengths, timestamps, MAC/SSID addresses) that are not compatible with standard ML algorithms. We thus use one - hot encoding to transform these non - numeric attributes into numerical ones, making the dataset ready for subsequent modeling. This preprocessing pipeline effectively improves data quality and computational efficiency for UAV applications.

3.2 Oversampling for data imbalance

Existing machine learning classifiers often perform well on balanced datasets, but real-world data distributions usually show significant class imbalance. In the operational context of WiFi-enabled UAVs, there is a severe class imbalance between abnormal and normal traffic data. Unprocessed data in this scenario tends to cause overfitting during training. Given the limited size of traffic data here, this study uses the Synthetic Minority Over-sampling Technique (SMOTE) [16] as the approach after comparing various sampling methods, with the implementation details provided in Equation (1).

$$x_{new} = x_i + \lambda \cdot (x_j - x_i) \tag{1}$$

Linear interpolation generates synthetic samples by randomly selecting points along the line segment connecting each original minority class sample *xi* and its *k* nearest neighbors. The position of the interpolated sample x_j is controlled by a random parameter $\lambda \in [0,1]$, which defines the weight between x_i and its neighbor. Equation (2) is employed to identify the *k* nearest neighbors for each minority sample, ensuring the interpolation process maintains the local topological structure of the minority class distribution.

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} (x_{i,k} - x_{j,k})^2} \tag{2}$$

To tackle data augmentation in UAV WiFi systems while preserving data integrity, we propose a framework combining SMOTE-based synthetic interpolation with iterative

decision tree evaluation. We oversample minority classes via SMOTE to expand datasets while maintaining feature distributions, using a lightweight decision tree to iteratively assess sample quality for efficiency and performance. Key innovations include tuning SMOTE's K-values (5–9) via a correlation metric linking feature dimensionality to training dynamics, determining K=7 as optimal to balance diversity and noise. Post-augmentation feature selection uses decision tree scores to refine datasets. Experiments show K=7 avoids underfitting (K<5) and noise (K>7), offering a reproducible method for efficient, generalizable UAV communication models.

3.3 Feature selection

In UAV communication systems, high-accuracy real-time intrusion detection is critical, but high-dimensional datasets strain machine learning models with heavy computational costs. Traditional Boruta-based methods [17] improve prediction via shadow features and iterative ranking, yet their resource-intensive cycles cause significant overhead and latency. We propose a lightweight framework that streamlines Boruta by optimizing shadow feature generation and applying hierarchical ranking [18]. This efficiently selects discriminative features, reduces dimensionality, and maintains accuracy while cutting computational complexity. Preprocessing datasets with our method enables resource-efficient inputs for downstream models, ensuring optimal real-time performance in UAV systems.

Variance Threshold Filtering

The Boruta algorithm generates shadow features and compares their importance with original ones. In high-dimensional data, doubling feature space raises computational costs due to random forest complexity. Many irrelevant features can make shadow features spuriously inflate scores, obscuring genuine relevant feature detection, while low-variance features add noise. Despite this, SMOTE oversampling during preprocessing boosts dataset size with synthetic samples, maintaining Boruta's robustness. Iterative shadow feature comparisons stably isolate critical features without manual bias.

For continuous features, standardization (e.g., Z-score) is vital before variance calculation to eliminate distortions from dimensional disparities. Variance relies on sample statistical dispersion, as defined in Equation (3).

$$\mathrm{Var}(t) = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})^2 \tag{3}$$

For features represented through one-hot encoding, the variance of a binary feature can be calculated using Equation (4):

$$Var(t) = p(1-p) \tag{4}$$

To streamline feature extraction and reduce computational overhead, a median-based variance thresholding method is adopted. This approach automatically selects features with variance above the median, eliminating the need for manual threshold adjustment

and reducing the number of features by half. It serves as an efficient initial dimensionality reduction step in dataset preprocessing, laying a solid groundwork for subsequent fine-grained feature engineering using the Boruta algorithm.

High-Correlation Feature Elimination

In the initial feature screening, the variance filtering method only measures individual feature dispersion but ignores their association with the target variable. To address this, we use the Pearson correlation coefficient [19] to eliminate irrelevant/redundant features, retain critical information, and reduce overfitting risks. This metric systematically assesses linear relationships by calculating the ratio of the covariance between a feature and the target to the product of their standard deviations, providing a robust measure of feature-target dependency. For paired sample data, the Pearson coefficient r is computed using formula (5):

$$r = \frac{\sum_{i=1}^{n}(x_i-\bar{x})(y_i-\bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i-\bar{x})^2}\sqrt{\sum_{i=1}^{n}((y_i-\bar{y})^2)}} \tag{5}$$

The correlation coefficient ranges between [-1, 1], where +1 indicates perfect positive linear correlation (variables change in the same direction) and -1 represents perfect negative linear correlation (variables change in opposite directions). Variables with a correlation coefficient absolute value exceeding 0.7 are identified as highly correlated to eliminate redundant variables in the UAV WiFi traffic dataset, address explicit collinearity issues, and enhance model robustness. The remaining variables undergo a two-step variance inflation factor [20] (VIF) verification to evaluate their linear dependence on other variables, resolve implicit multicollinearity, and ensure all VIF values remain below 5. The VIF formula quantifies the degree to which a variable's variance is inflated due to multicollinearity in the dataset. The variance inflation factor (VIF) is calculated using the formula (6):

$$VIF_j = \frac{1}{1-{R_j}^2} \tag{6}$$

In the VIF formula, $1 - {R_j}^2$ measures the unexplained variance proportion of the j-th predictor by other variables. VIF quantifies how much multicollinearity magnifies the j-th coefficient's variance versus a collinearity-free model. Iteratively removing high-VIF variables and applying Boruta for feature importance ensures a stable model with critical features retained.

Feature importance selection using Boruta algorithm

In the initial phase, variance thresholding removes low-variance features. Pearson correlation and VIF iteratively eliminate collinear variables. The Boruta algorithm identifies predictive features by capturing nonlinear relationships, systematically removing redundancies while preserving critical patterns.

Boruta, a Random Forest (RF)-based method, generates shadow features for each original variable and compares their importance scores. Features with consistently

higher importance than their shadows are retained; others are discarded, ensuring robust selection of uniquely contributive features.

For an original feature matrix $X \in \mathbb{R}^{n\times m}$, each feature X_j generates a shadow feature X_j^{shadow}, with noise-independent surrogates serving as a control group. In classification tasks, feature importance is assessed via Gini impurity reduction during RF splits, with scores standardized using formula (7).

$$Z_j = \frac{Importance_j - \mu_{shadow}}{\sigma_{shadow}} \tag{7}$$

μ_{shadow} and σ_{shadow} denote the mean and standard deviation, respectively, of the importance scores for all shadow features.

To prioritize original over shadow features, the critical threshold is set as $\mathrm{Z_{max}} = \max(\mathrm{Z^{shadow}})$. Features with $Z_j < Z_{max}$ are dynamically pruned in each iteration to avoid loops and boost efficiency, while remaining features generate new shadow features for subsequent rounds. The algorithm stops when feature states remain unchanged for three consecutive iterations or after 100 iterations, outputting importance scores and history data, and identifying the top feature as visualized.

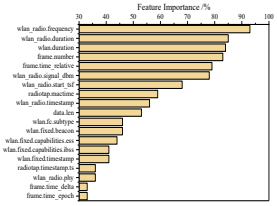

Fig. 3. Feature Importance of selected features

3.4 Attack Classification with Modified Loss Function

The benchmark model in this method, Gradient Boosting Decision Tree (GBDT), is an iterative ensemble model aggregating weak learners to generate classification results. By residual transfer and gradient fitting, it uses the previous tree's output as the next input. Residual calculation, the core iterative step in GBDT, directly affects training speed. The proposed optimization accelerates it: simplifying loss function computation significantly reduces the model's training time.

As enhancements, XGBoost adds a regularization term to the objective function to address GBDT's overfitting issue. LightGBM, another improved variant, boosts training efficiency via Gradient-based One-Side Sampling (GOSS), simplifies high-dimensional features with Exclusive Feature Bundling (EFB), and adopts a leaf-wise tree growth strategy to reduce computational complexity compared to XGBoost's level-wise approach.

In WiFi drone intrusion detection, real-time demands are extremely high. The traditional GBDT [21] model uses a logarithmic loss function (Formula 8), while XGBoost dynamically calculates second-order derivatives during loss computation. Both fail to meet millisecond-level response requirements.

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N}[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)] \quad (8)$$

In real - time scenarios with limited computational resources on edge devices, cross - entropy replaces the logarithmic function as the loss function. Its fast convergence cuts iterations, saving resources and time for forward propagation and gradient calculation. Yet, standard cross - entropy uniformly computes losses, letting majority [22] - class simple samples dominate gradient updates and hindering minority - class learning. In severe imbalance, majority - class gradient accumulation masks the minority - class optimization direction. Weighted cross - entropy assigns class - specific weights to tackle imbalance and improve minority - class recognition. Its loss formula is (9):

$$\text{H} = -\frac{1}{N}\sum_{i=1}^{N}[w_1 \cdot y_i log(p_i) + w_0 \cdot (1-y_i) log(1-p_i)] \quad (9)$$

In this formulation, y_i represents the binary true label (0 or 1), p_i denotes the predicted probability of the positive class. For classifying rare abnormal classes in intrusion detection, we assign higher weights to low-frequency samples to balance data distribution. We adopt a computationally efficient class reciprocal method: $w_i = \frac{1}{N_i}$. In our hijacked traffic dataset, abnormal traffic samples constitute 20% of the total, so their initial weight is set to 5, with weights dynamically tuned based on experimental results.

Algorithm 1 briefly introduces the training process of the model.

Algorithm 1: Algorithm for Light-WIDS

Input: Train Data $\{(x_i, y_i)\}$ i=1 to N

Output: Light-WIDS Trained Model $\hat{y}_i^{(t)}$

1: Initialize the first tree as a constant:

$$\hat{y}_i^{(0)} = f_0 = 0$$

2: **while** t<maxRuns or $L_{(t)} < \epsilon$ **do**

3: Train the next tree by the loss of weighted cross entropy:

4: $f_t(x_i) = \arg_{f_t} wce\ L_{(t)}$

5: $f_t(x_i) = arg_{f_t} wce\ L\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right)$

6: Get the Next Model:

7: $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t(x_i)$

8: **end while**

9: Get Final Trained Model:

10: $\hat{y}_i^{(t)} = \sum_{t=0}^{M-1} f_t(x_i)$

11: **return** $\hat{y}_i^{(t)}$

At the start of the algorithm, the model initializes the first tree with a constant value for backward training. A stopping criterion is defined based on either reaching the maximum number of trees or reducing the loss below a threshold. Subsequent trees are trained using a weighted cross-entropy loss function. The iterative training process updates the model predictions through sequential tree additions, where:

- $\hat{y}_i^{(t)}$represents the prediction for sample i at iteration t
- $f_t(x_i)$ is the decision function of the t-th tree
- $L(t)=\sum(\hat{y}_i^{(t)} - y_i^{(t)})$ is the loss at iteration t
- ϵ is the minimum error threshold

Training proceeds for M iterations or until $L(t) < \epsilon$, returning the final ensemble model after satisfying the stopping condition.

4 Experiments

4.1 Experimental Environment and Evaluation Metrics

To validate the effectiveness of the proposed intrusion detection method, a series of experiments were designed and verified using a WiFi drone hijacking attack dataset collected through the following hardware configuration: (a) Parrot ANAFI Ai, (b) DJI Neo, (c) Raspberry Pi 4, (d) WiFi Pineapple Nano, (e) Alfa AWUS036NHA, alongside a laptop running an Ubuntu-based operating system. The Raspberry Pi 4 serves as the automated attack platform, while the WiFi Pineapple Nano both analyzes drone communications and participates in automated attacks. The AWUS036NHA supports monitoring mode and packet injection while being compatible with penetration testing systems like Kali Linux. The flow chart is presented below.

Once the Parrot drone is powered on, it will activate its WiFi access point (AP), with the remote controller functioning as the client. Use Vis tumbler to identify both the WiFi channel and the AP MAC address of the drone. Locate the WiFi network starting with "ANAFI-" and record the corresponding MAC address of the drone's WiFi signal.

Start Alfa AWUS036NHA in monitor mode to scan 2.4GHz for active channels/drone MACs. Find the MAC address of the drone's connected remote controller using its active WiFi address.

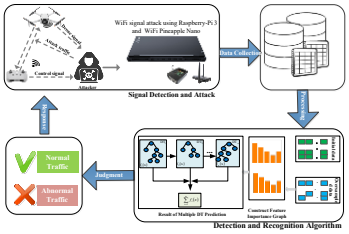

Fig. 4. UAV WiFi traffic collection and detection

Use aireplay-ng to conduct an attack on the remote control and capture packets on specific channels with Wireshark. Extract MACs (SA/DA/TA/RA), frame types, packet length, timestamp, duration from captured S802.11 packets. Group by SA, compute TN, AL, RF, RD per unit time. Build traffic dataset for ML input.

This dataset encompasses representative WiFi drone hijacking attacks [23], with detailed descriptions provided below.

1)Deauth Attack: To hijack drone control, attackers DE authenticate the pilot, forcing the drone to stop. A rogue controller (running the official app) then reconnects immediately.

2)MITM: Use WiFi Pineapple Nano to perform man-in-the-middle attacks (MITM) on drones that support WiFi control, mainly by forging an SSID with the same name as the target drone, inducing it to connect to the forged malicious AP.

3)Kr00K: The attacker used Air crack-ng to send fake disassociation frames to the drone, causing it to repeatedly disconnect and reconnect. During this process, the attacker captured encrypted data packets via a wireless network card and decrypted them.

4)DoS: By launching a simple ping flood using hping3 without waiting for any replies, the target was inundated with requests, rendering it unable to make any other communications.

5) (Re)Assoc: After a DE authentication attack severs the drone-ground station link, Aircrack-ng floods the target AP with high-density association requests, blocking legitimate connections.

6)KRACK Attack: Attackers replay the third handshake message during the process, forcing the drone to reinstall an already used key. This directly disrupts the protocol mechanism, enabling decryption of traffic without obtaining the WiFi password.

Table 1 presents the composition of the drone WiFi hijacking traffic dataset, detailing attack types alongside the counts of normal and malicious traffic packets.

Table 1. Dataset Information

Attack Type	Normal Traffic	Attack Traffic	Total Traffic
Deauth	9705	1487	11192
MITM	5797	313	6110
Kr00k	2873	683	3556
DoS	2468	288	2756
(Re)Assoc	1843	155	1998
KRACK	1349	249	1598

This dataset focuses on the most severe drone WiFi hijacking attacks, generating attack data through 2-second bursts of malicious packets every 20 seconds.

To evaluate WiFi Intrusion Detection Systems (WiFi IDS), we use a comprehensive metric framework covering both detection accuracy and operational efficiency. Precision and recall quantify the system's ability to correctly identify attacks and classify legitimate traffic: true positives (TP) denote accurate attack detections, while true negatives (TN) represent correct normal traffic classifications. False positives (FP) signify misclassifying legitimate traffic as attacks, and false negatives (FN) indicate undetected actual attacks.

Supplementing these accuracy metrics are timing measurements: training time measures model-building duration, test time assesses dataset validation efficiency, and inference time gauges real-time processing speed. This integrated approach ensures a balanced evaluation of detection reliability and system responsiveness for practical deployments.

Accuracy is the number of successfully predicted samples to the total number of samples.

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{10}$$

Recall R is the detection rate or true positive rate (TPR).

$$Recall = \frac{TP}{TP+FN} \tag{11}$$

Training time T_a defines how much time a method is utilized to train and build the model using the whole training dataset.

$$T_a = End_{train\_time} - Start_{train\_time} \tag{12}$$

Testing time T_b specifies the amount of time it takes for a method to predict the full testing dataset.

$$T_b = End_{test\_time} - Start_{test\_time} \tag{13}$$

Inference time T_c refers to the time taken by a method to process the entire reasoning dataset through its inference process.

$$T_c = End_{infer\_time} - Start_{infer\_time} \tag{14}$$

4.2 Comparison Experiments on Our dataset

Multi-Attack Detection

Given the frequent hybrid attack scenarios in drone WiFi communications, assessing the detection performance of the proposed method against multiple hybrid attacks is essential.

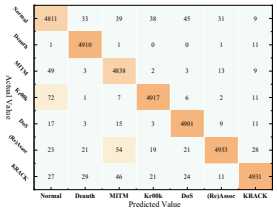

Fig. 5. Multi-classification confusion matrix

Table 2. Multiple attack detection results

Attack Type	Accuracy	Recall	F1
Normal	0.9541	0.9421	0.9667
Deauth	0.9814	0.9765	0.9721
MITM	0.9513	0.9497	0.9536
Kr00k	0.9843	0.9827	0.9852
DoS	0.9767	0.9675	0.9777
(Re)Assoc	0.9911	0.9865	0.9873
KRACK	0.9901	0.9887	0.9897

The confusion matrix results of multi-class classification between predicted and true labels offer a comprehensive assessment of each category's classification performance. The method demonstrates excellent detection performance for normal data and various attack types. Notably, under mixed attack scenarios, the overall detection accuracy reaches 96.3%, validating the proposed intrusion detection system's ability to effectively identify attacks in complex environments with simultaneous multiple threats and highlighting its robustness in real-world applications.

Comparison with Traditional Algorithms

This paper evaluates multiple classical machine learning algorithms (DT[24], RF[24], XGBoost[25]) for detecting drone WiFi traffic attack datasets. The accuracy comparisons in the figure demonstrate that among traditional models, our proposed method and XGBoost emerge as top performers. Notably, our approach achieves the highest accuracy of 96.3%. These results validate the feasibility of our method, which demonstrates superior accuracy and robustness across diverse attack types.

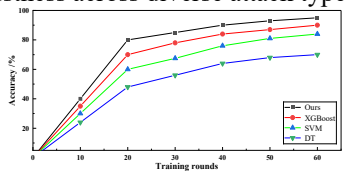

Fig. 6. Comparative analysis of traditional ML algorithms using accuracy metrics

In practical scenarios, addressing strict real-time constraints for drone WiFi traffic detection requires prompt responses to malicious WiFi attacks. We employ a 5:2:3 split for training, testing, and inference validation sets. Comparative experiments are conducted with four classic algorithms under identical conditions, evaluating their real-time performance across training, testing, and inference times.

Table 3. Time consumption comparison results table

Model	Training time	Test time	Inference time
DT	201	110	177
SVM	830	100	198
RF	460	100	120
XGBoost	1669	70	88
Ours	181	51	71

The table shows that traditional models have similar test and inference performance under the same conditions, but their training times differ due to algorithm complexity. XGBoost achieves efficient testing and inference via gradient boosting and hyperparameter tuning, though it requires significantly longer training than other models. Our method, which adds a feature selection module and modifies the loss function, outperforms all baselines in training, testing, and inference. Notably, it cuts training time by 89% compared to XGBoost, demonstrating the effectiveness of our feature selection and weighted cross-entropy loss in improving computational efficiency.

4.3 Comparison Experiment on Lightweight of Algorithm

This study evaluates WiFi drone intrusion detection algorithms under real-world operational constraints, emphasizing high real-time requirements and limited onboard computing resources. The comparison framework assesses algorithms across five dimensions: detection accuracy, recall rate, computational complexity, parameter count, and model size - ensuring a comprehensive balance between detection performance and operational feasibility for resource-constrained UAV systems.

Table 4. Algorithm performance comparison experiment

Model	Accuracy /%	Recall /%	Parameters /10^6	FLOPs(G)	Size(M)
DT	88.1	82.3	2.6	2.9	3.1
SVM	92.2	87.4	4.5	3.5	6.2
RF	94.5	89.3	14.5	7.8	15.2
XGBoost	95.1	90.6	6.7	5.4	8.3
FED-IDS	95.5	91.3	32	10.6	43
CGAN-IDS	96.6	92.4	41	13.3	56
Ours	96.3	93.5	0.9	1.3	1.5

Results show this model achieves 96.3% accuracy/93.5% recall (2nd to CGAN-IDS), outperforming DT/SVM [24]. While matching CGAN-IDS' performance [13], it uses 35× fewer parameters and 10× less compute, enabling cost-effective deployment. Traditional models lag in accuracy/recall despite lower parameters. Ideal for lightweight scenarios like low-power drones, balancing high performance with minimal resource use.

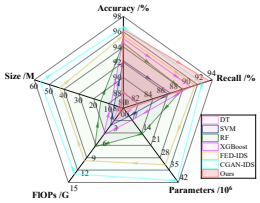

Fig. 8. Comparative experiment on lightness

4.4 Ablation Experiments

To systematically evaluate the contribution of each enhanced component to the algorithm's overall performance and validate the cumulative improvements, an ablation study was designed. This experiment systematically isolated the impact of three key modules: 1) an optimized data preprocessing pipeline, 2) a novel lightweight feature selection mechanism, and 3) a weighted cross-entropy loss function replacing the original loss function. By analyzing performance across different module combinations, this study quantifies the individual impacts of each optimization and reveals their synergistic effects on detection accuracy and computational efficiency.

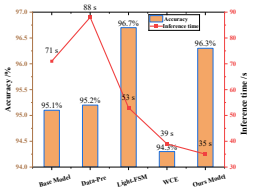

Fig.9. Compare the various improvement modules

Ablation studies showed that optimized data preprocessing slightly increased inference latency but significantly boosted detection accuracy. Other modules delivered mixed improvements in accuracy or efficiency. The fully integrated model achieved 96.3% top accuracy (+1.2% over baseline) with a minimized inference time of 35 seconds (50% faster than the original design). The Light-FSM and WCE modules demonstrated strong synergy: Light-FSM enhanced detection precision, while WCE effectively reduced computational costs. These results validate the proposed modules' effectiveness in detecting drone-based WiFi hijacking intrusions through complementary performance gains.

5 Conclusions

This paper introduces a lightweight real-time detection algorithm for drone WiFi intrusion and hijacking attacks, tailored for drones with limited computing resources and strict real-time communication demands. Experimental results show that the proposed model surpasses traditional machine learning methods in detection accuracy. Compared with high-accuracy deep learning-based intrusion detection models, it achieves superior efficiency with significantly lower computational overhead. The optimized feature selection mechanism and simplified loss function enhance time efficiency, ensuring real-time traffic analysis capabilities. This work confirms the feasibility of the proposed method for drone WiFi security, providing a novel solution to protect unmanned aerial vehicle (UAV) communication systems.

Future research will explore other challenges in UAV communication scenarios, focusing on improving the model's generalization ability and integrating dynamic learning algorithms to address emerging network threats.

Acknowledgments. This study was funded by National Natural Science Foundation of China (U2333201) and the fundamental research funds for the central universities (3122025094).

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Huang, C., Fang, S., Wu, H., Wang, Y., & Yang, Y.: Low-altitude intelligent transportation: System architecture, infrastructure, and key technologies. Journal of Industrial Information Integration, 42, 100694(2024)
2. Zhou, Y.: Unmanned Aerial Vehicles based low-altitude economy with lifecycle techno-economic-environmental analysis for Sustainable and Smart Cities. Journal of Cleaner Production, 145050 (2025)
3. Rugo, A., Ardagna, C. A., & Ioini, N. E.: A security review in the UAVNet era: Threats, countermeasures, and gap analysis. ACM Computing Surveys, 55(1), 1-35(2022)
4. Wang, Z., Li, Y., Wu, S., Zhou, Y., Yang, L., Xu, Y., ... & Pan, Q.: A survey on cybersecurity attacks and defenses for unmanned aerial systems. Journal of Systems Architecture, 138, 102870(2023)
5. Lee, A. Y. P., Wang, M. I. C., Hung, C. H., & Wen, C. H. P.: PS-IPS: Deploying Intrusion Prevention System with machine learning on programmable switch. Future Generation Computer Systems, 152, 333-342(2024)
6. Yaacoub, J. P., Noura, H., Salman, O., & Chehab, A.: Security analysis of drones systems: Attacks, limitations, and recommendations. Internet of Things, 11, 100218 (2020)
7. Strohmeier, M., Lenders, V., Martinovic, I.: Intrusion Detection for Airborne Communication Using PHY-Layer Information. In: Almgren, M., Gulisano, V., Maggi, F. (eds) Detection of Intrusions and Malware, and Vulnerability Assessment. DIMVA 2015. LNCS, vol 9148, pp. 67-77. Springer, Cham (2015)
8. Condomines, J. P., Zhang, R., & Larrieu, N.: Network intrusion detection system for UAV ad-hoc communication: From methodology design to real test validation. Ad Hoc Networks, 90, 101759 (2019)
9. Zang, M., & Yan, Y.: Machine learning-based intrusion detection system for big data analytics in VANET. In 2021 IEEE 93rd vehicular technology conference, pp. 1-5. IEEE, Piscataway (2021)
10. Das, A.: Design and development of an efficient network intrusion detection system using ensemble machine learning techniques for WiFi environments. International Journal of Advanced Computer Science and Applications, 13(4) (2022)
11. Ahmad, U. B., Akram, M. A., & Mian, A. N.: Low-latency intrusion detection using a deep neural network. IT Professional, 24(3), 67-72 (2022)
12. Sayegh, H. R., Dong, W., & Al-madani, A. M.: Enhanced intrusion detection with LSTM-Based model, feature selection, and SMOTE for imbalanced data. Applied Sciences, 14(2), 479(2024)
13. Tabassum, A., Erbad, A., Lebda, W., Mohamed, A., & Guizani, M.: Fedgan-ids: Privacy-preserving ids using gan and federated learning. Computer Communications, 192, 299-310(2022)
14. Thing, V. L.: IEEE 802.11 network anomaly detection and attack classification: A deep learning approach. In 2017 IEEE wireless communications and networking conference, pp. 1–6. IEEE, Piscataway (2017)
15. Ran, J., Ji, Y., & Tang, B.: A semi-supervised learning approach to IEEE 802.11 network anomaly detection. In 2019 IEEE 89th vehicular technology conference, pp. 1–5. IEEE, Piscataway (2019)
16. Arafa, A., El-Fishawy, N., Badawy, M., & Radad, M.: RN-SMOTE: Reduced Noise SMOTE based on DBSCAN for enhancing imbalanced data classification. Journal of King Saud University-Computer and Information Sciences, 34(8), 5059-5074 (2022)

17. Liu, X., Li, Z., Zhang, Z., Li, S., & Zhang, G.: Integrated framework for EMD–Boruta-LDA feature extraction and SVM classification in coal and gas outbursts. Journal of Experimental & Theoretical Artificial Intelligence, 35(8), 1121-1140 (2023)
18. Chen, J., Ying, Z., Zhang, C., & Balezentis, T.: Forecasting tourism demand with search engine data: A hybrid CNN-BiLSTM model based on Boruta feature selection. Information Processing & Management, 61(3), 103699(2024)
19. Liu, P., Wang, S., & Zhao, P.: Robust estimation and test for Pearson's correlation coefficient. Random Matrices: Theory and Applications, 13(04), 2450023(2024)
20. Salmerón-Gómez, R., García-García, C. B., & García-Pérez, J.: A redefined variance inflation factor: overcoming the limitations of the variance inflation factor. Computational Economics, 1-27(2024)
21. Sun, S., & Wang, J.: Ship Detection in SAR Images Based on Steady CFAR Detector and Knowledge-Oriented GBDT Classifier. Electronics, 13(14), 2692 (2024)
22. Li, Y., Jiao, Z., Wang, S., Feng, K., & Liu, Z.: Cross diversity entropy-based feature extraction for fault diagnosis of rotor system. IEEE/ASME Transactions on Mechatronics, 29(3), 1831-1843 (2023)
23. Westerlund, O., & Asif, R.: Drone hacking with raspberry-pi 3 and wifi pineapple: Security and privacy threats for the internet-of-things. In 2019 1st International Conference on Unmanned Vehicle Systems-Oman, pp. 1-10. IEEE, Piscataway (2019, February)
24. Mohiuddin, G., Lin, Z., Zheng, J., Wu, J., Li, W., Fang, Y., ... & Zeng, X.: Intrusion detection using hybridized meta-heuristic techniques with Weighted XGBoost Classifier. Expert Systems with Applications, 232, 120596(2023)
25. Sagi, O., & Rokach, L.: Approximating XGBoost with an interpretable decision tree. Information sciences, 572, 522-542 (2021)