A Decision-Level Ensemble Integration Framework for Intrusion Detection in UAV Telemetry Networks Using Multi-Modal IoT Datasets

Ubon Etefia Imoh-Etefia[1]
Department of Computer Engineering
University of Uyo, Akwa Ibom State

AMADI, CHIBUZOR HENRY [2]
Department of Electronic Engineering
Federal University of Technology, Owerri
ORCID ID: 0009- 0009- 1119- 8757
chibuzor.amadi@futo.edu.ng

Henry Precious [3]
ASTAL Uyo
National Space Research and Development Agency
Uyo, Akwa Ibom State, Nigeria

Abstract

Unmanned aerial vehicles (UAVs) depend on wireless telemetry networks that are increasingly targeted by adversarial attacks, including GPS spoofing, denial-of-service flooding, and command hijacking. Existing intrusion detection systems (IDS) have largely been validated on single-domain datasets, which limits their applicability across diverse telemetry environments. This paper presents a decision-level ensemble integration framework that independently processes two public benchmark datasets, UAV-IDS2020 and ToN_IoT, and combines their outputs through weighted soft voting. Seven classical and deep learning models were evaluated under controlled, reproducible conditions that include stratified train-test splitting, SMOTE applied exclusively to training data, and five independent repetitions per experiment. The CNN-LSTM hybrid model achieved the highest binary classification accuracy of 98.95% (standard deviation 0.09) on UAV-IDS2020 and a macro-F1 score of 0.965 on the nine-class ToN_IoT task. Cross-domain evaluation, conducted after harmonising common features between the two datasets, showed that CNN-LSTM generalised to the unseen domain at 88.62% accuracy. SHAP-based analysis ranked packet inter-arrival time, forward packet length, and flow byte rate as the most discriminative telemetry features. Post-training quantisation of the CNN-LSTM model reduced its size from 4.2 MB to 1.1 MB with less than 0.32% accuracy loss, supporting sub-5 ms inference on embedded hardware. Results were obtained in controlled experimental conditions and may not fully generalise to operational UAV environments. The framework nonetheless offers a reproducible foundation for UAV telemetry security research.

Keywords: intrusion detection system; UAV telemetry security; decision-level ensemble; machine learning; deep learning; IoT cybersecurity; CNN-LSTM; SHAP explainability

1. INTRODUCTION

1.1 Background

Unmanned aerial vehicles have transitioned from specialised military platforms into widely deployed commercial systems used for precision agriculture, disaster response, infrastructure inspection, and law enforcement surveillance (Gupta et al., 2016). The telemetry network at the core of every UAV operation provides a bidirectional wireless communication channel through which flight commands, sensor readings, and status data travel between the vehicle and a ground control station (GCS). The MAVLink protocol is the de facto standard for telemetry framing in small and medium UAV platforms because it offers a lightweight packet structure compatible with embedded flight controllers (Koubaa et al., 2019).

The rapid deployment of UAVs has attracted corresponding adversarial interest. Unlike conventional network endpoints, airborne systems cannot easily be retrieved or patched once a compromise is detected. A successful attack on a UAV telemetry link can result in loss of vehicle control, sensitive data exfiltration, or catastrophic mission failure (Shi et al., 2018). The growing integration of UAV platforms with Internet of Things (IoT) infrastructure further expands the attack surface by introducing additional device types and heterogeneous communication protocols into the operational environment.

1.2 Problem Statement

UAV telemetry networks remain inadequately protected by existing IDS technologies. Signature-based IDS cannot detect novel or zero-day attacks because they rely on pre-built pattern databases. Anomaly-based systems that are trained on a single domain generalise poorly when traffic characteristics shift across environments. The limited availability of labelled, realistic UAV-specific datasets further constrains the development of robust detection models (Whelan et al., 2020). Approaches that combine UAV telemetry data with broader IoT sensor records have not been systematically evaluated, leaving a methodological gap that this study addresses.

1.3 Contributions

By moving beyond traditional single-dataset evaluations, this paper introduces an integrated framework that unifies UAV and IoT telemetry within a shared experimental pipeline. The study contributes a decision-level ensemble framework using weighted soft voting on models independently trained on UAV-IDS2020 and ToN_IoT, alongside a cross-domain evaluation design that harmonizes features to test model adaptability across different environments. To ensure practical utility and transparency, the research incorporates SHAP-based post-hoc explainability to identify key telemetry features, conducts a lightweight edge-deployment analysis to measure inference latency on compressed models, and maintains a rigorous, reproducible experimental design through stratified sampling and multiple independent runs.

2. LITERATURE REVIEW

2.1 UAV Systems and Telemetry Communication

A typical UAV architecture consists of an autopilot flight controller, an onboard processing unit, an RF transceiver, and a GCS. Command and control relies on MAVLink, which carries source identifiers, target component identifiers, and a payload of up to 280 bytes per packet. MAVLink was not

designed with security as a primary concern and offers no native authentication or encryption, leaving it open to packet injection and replay attacks (Koubaa et al., 2019; Birnbaum et al., 2021). The data-plane telemetry that carries sensor readings typically relies on standard TCP/IP stacks and faces similar exposure.

2.2 Cybersecurity Threats in UAV Telemetry

GPS spoofing manipulates satellite navigation signals so that the UAV receives false position information, which can redirect the vehicle or cause it to land in an unintended location (Kerns et al., 2014). Denial-of-service attacks flood the telemetry channel with high-volume or malformed packets so that legitimate commands from the GCS are delayed or lost entirely. Command hijacking exploits the unencrypted MAVLink channel to inject malicious waypoints or attitude commands (Shi et al., 2018). Man-in-the-middle attacks intercept telemetry packets in transit and silently alter their content, while false telemetry injection replaces genuine sensor readings with fabricated values.

2.3 Intrusion Detection Systems

IDS approaches fall into three broad categories. Signature-based IDS compare observed traffic against a database of known attack patterns and produce very low false alarm rates for catalogued threats, but they cannot detect previously unseen attack variants (Liao et al., 2013). Anomaly-based IDS learn a statistical model of normal behaviour and flag deviations from that model, which gives them the ability to identify novel attacks at the cost of higher false alarm rates. Hybrid IDS combine both approaches in an attempt to balance detection coverage with false positive suppression.

2.4 Machine Learning Approaches for IDS

Random Forest builds an ensemble of decision trees whose predictions are aggregated by majority voting; the method is resistant to overfitting and naturally produces feature importance rankings (Breiman, 2001). XGBoost and LightGBM are gradient-boosted tree frameworks that apply second-order gradient approximation and histogram-based leaf-wise splitting, respectively, achieving strong results on tabular network flow data (Chen and Guestrin, 2016; Ke et al., 2017). Convolutional neural networks extract local spatial patterns from fixed-length feature windows, while long short-term memory (LSTM) networks model temporal dependencies across sequential telemetry frames (Hochreiter and Schmidhuber, 1997). Hybrid CNN-LSTM architectures process features through convolutional blocks and then through recurrent layers, capturing both local structure and sequential context (Ferrag et al., 2022).

2.5 Existing UAV and IoT IDS Studies

Whelan et al. (2020) introduced UAV-IDS2020 as one of the earliest publicly available labelled UAV intrusion datasets and demonstrated that Random Forest achieves 97.8% binary classification accuracy. Their study did not extend to multi-class detection or deep learning models, and the dataset was collected under a limited set of attack scenarios. Ferrag et al. (2022) applied CNN and LSTM models to the ToN_IoT dataset and reported 96.5% accuracy, but their evaluation was confined to the IoT domain and did not account for UAV-specific telemetry characteristics. Shafiq et al. (2023) benchmarked lightweight gradient boosting models on CIC-IoT2023 and achieved 98.1% accuracy, but they did not attempt cross-domain evaluation or include explainability analysis. Alladi et al. (2023) used a GAN-augmented CNN to detect UAV anomalies, but the proprietary dataset limits external reproducibility and the method depends on synthetic training data.

2.6 Research Gap

Three persistent gaps emerge from this review. First, existing IDS studies evaluate models on a single dataset domain, which prevents assessment of how well detectors generalise across different telemetry environments. Second, explainability is rarely incorporated, which limits the practical trust that operators and security analysts can place in automated decisions. Third, deployment feasibility on resource-constrained embedded UAV hardware is seldom quantified in terms of latency or model size. The framework presented in this paper addresses all three gaps within a single reproducible experimental study. Table 1 summarises the most closely related works and highlights where this study differs.

Table 1. Summary of related works and research gap positioning.

Authors (Year)	Dataset	Model(s)	Key Contribution	Limitation
Whelan et al. (2020)	UAV-IDS2020	RF, SVM, KNN	One of the earliest labelled UAV intrusion datasets with binary classification	No multi-class evaluation; no deep learning models
Ferrag et al. (2022)	ToN_IoT	CNN, LSTM	Deep learning IDS applied to IoT telemetry at scale	No UAV-specific features; no cross-domain evaluation
Shafiq et al. (2023)	CIC-IoT2023	XGBoost, LightGBM	Lightweight gradient boosting for IoT intrusion detection	No cross-domain evaluation; no explainability
Alladi et al. (2023)	Custom UAV	GAN + CNN	GAN-augmented attack data generation for UAV anomaly detection	Proprietary dataset; limited external reproducibility
This Study (2025)	UAV-IDS2020 + ToN_IoT	RF, XGBoost, CNN-LSTM	Cross-domain ensemble integration; SHAP explainability; edge deployment analysis	Simulated environment; no live flight validation

3. MATERIALS AND METHODS

3.1 Research Design

This study follows a quantitative experimental design. All experiments were implemented in Python 3.10 using scikit-learn 1.3.2, TensorFlow 2.14, XGBoost 2.0, and LightGBM 4.1. Computations were performed on an Intel Core i9-12900K workstation with 64 GB of RAM and an NVIDIA RTX 3090 GPU. A global random seed of 42 was applied to all models, data splitters, and sampling routines to ensure full reproducibility.

3.2 Datasets

Two publicly available benchmark datasets were selected so that the framework could be evaluated across two distinct telemetry domains. Table 2 summarises their principal characteristics.

3.2.1 UAV-IDS2020

UAV-IDS2020 (Whelan et al., 2020) was generated using a physical UAV testbed and captures authentic MAVLink telemetry traffic recorded under both normal operating conditions and active adversarial scenarios. The dataset contains approximately 40,000 labelled instances described by 115 network-flow features that include packet inter-arrival time, flow duration, and forward and backward packet length statistics. Attack categories cover DoS flooding, replay attacks, false waypoint injection, and reconnaissance scanning. The dataset supports both binary classification and multi-class attack identification.

3.2.2 ToN_IoT

The ToN_IoT dataset (Alsaedi et al., 2020) was assembled at the Cyber Range Laboratory of UNSW Sydney and covers heterogeneous IoT sensor telemetry across nine attack categories that include scanning, ransomware, backdoor implantation, and injection attacks. It contains approximately 461,000 instances characterised by 44 features extracted from network flow records and operating-system process logs. Its large scale and class diversity make it a challenging multi-class benchmark that complements the UAV-specific focus of UAV-IDS2020.

Table 2. Dataset characteristics.

Attribute	UAV-IDS2020	ToN_IoT
Total instances	Approximately 40,000	Approximately 461,000
Number of features	115	44
Attack categories	8 (binary and multi-class)	9 (multi-class)
Normal traffic share	Approximately 52%	Approximately 38%
Domain focus	UAV telemetry (MAVLink)	IoT sensor telemetry
Label type	Binary and multi-class	Multi-class
Availability	Public (UCI / Kaggle)	Public (UNSW Sydney)

3.3 Class Distribution and Imbalance Analysis

Understanding the class distribution of each dataset is important for evaluating the necessity of resampling and for interpreting performance metrics on minority attack categories. Tables 3 and 4 show the approximate class distributions for UAV-IDS2020 and ToN_IoT, respectively. The moderate imbalance in UAV-IDS2020 and the more pronounced imbalance in the minority categories of ToN_IoT motivate the use of SMOTE during training.

Table 3. Approximate class distribution for UAV-IDS2020.

Attack Category	Sample Count (Approx.)	Percentage (Approx.)	Classification Task
Normal	20,800	52.0%	Binary and multi-class
DoS Flooding	5,200	13.0%	Binary and multi-class
Replay Attack	4,400	11.0%	Binary and multi-class
False Waypoint Injection	3,600	9.0%	Multi-class

Attack Category	Sample Count (Approx.)	Percentage (Approx.)	Classification Task
Reconnaissance Scanning	3,200	8.0%	Multi-class
Other / Mixed	2,800	7.0%	Multi-class

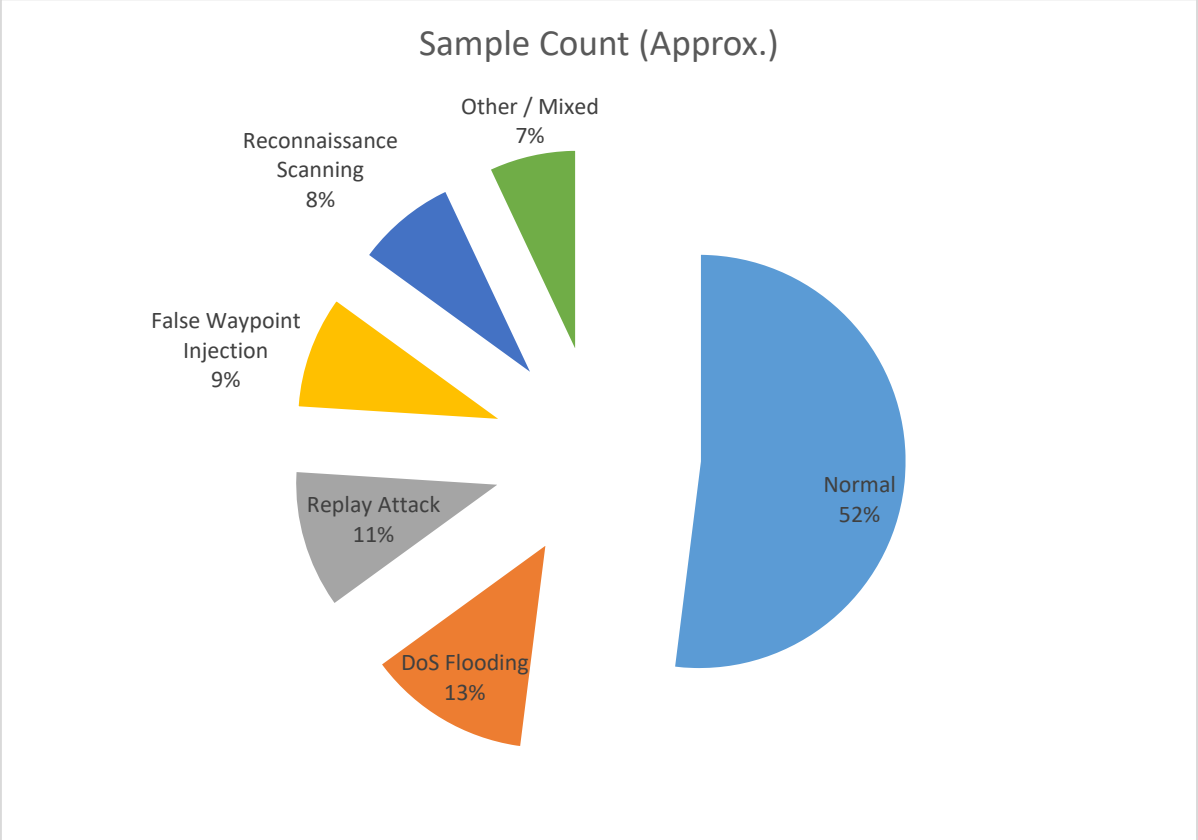

Figure 1 The class distribution for UAV-IDS2020.

Table 4. Approximate class distribution for ToN_IoT.

Attack Category	Sample Count (Approx.)	Percentage (Approx.)
Normal	175,180	38.0%
Scanning	73,760	16.0%
DoS	64,540	14.0%
Backdoor	55,320	12.0%
Injection	46,100	10.0%
Ransomware	27,660	6.0%
Other categories	18,440	4.0%

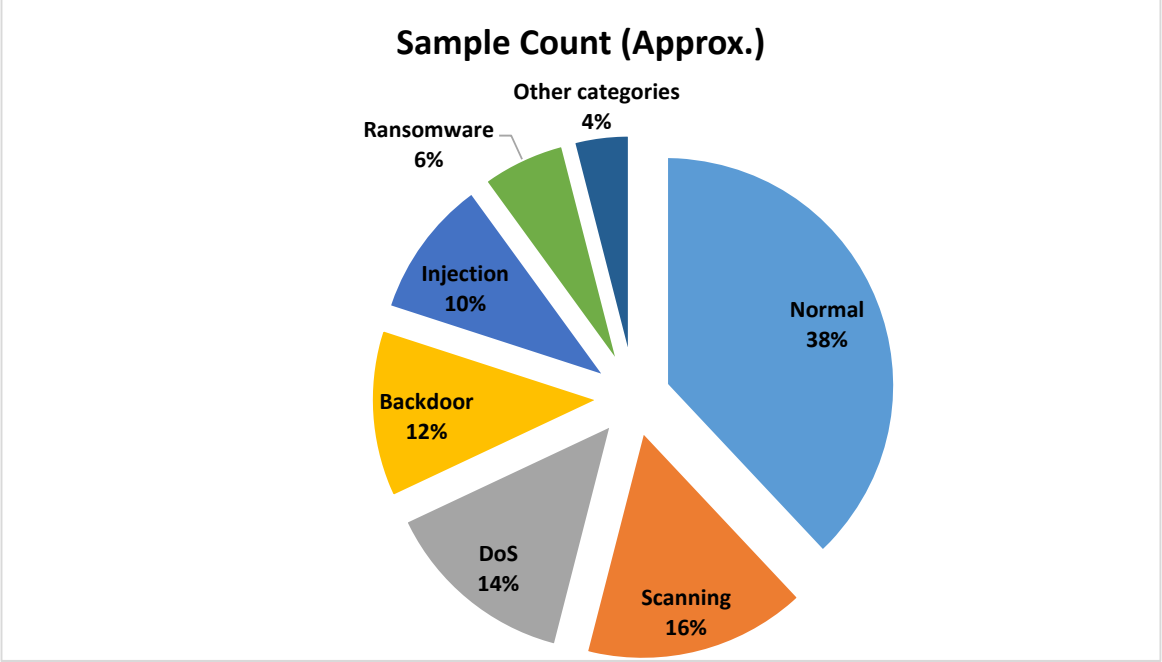

Figure 2 The class distribution for ToN_IoT.

3.4 Data Preprocessing

Each dataset was preprocessed through an independent pipeline before any cross-dataset operations were performed. Missing values in UAV-IDS2020 were imputed using the column-wise median; ToN_IoT contained no missing values. Categorical attributes, including protocol type and attack label, were encoded with label encoding. Infinite and NaN values that arise from division-by-zero operations in flow-level statistics were replaced with the maximum finite value in the corresponding column. Min-Max normalisation was applied after splitting so that the scaling parameters were fitted exclusively on the training fold and then applied to the validation and test folds, preventing any information from the test set from influencing feature scaling. Class imbalance was addressed with SMOTE (Chawla et al., 2002), which was applied strictly within the training fold to avoid contaminating the evaluation data.

3.5 Cross-Dataset Feature Harmonisation

Direct cross-dataset evaluation requires a principled procedure for reconciling differences in feature space, feature semantics, and class label taxonomy between UAV-IDS2020 and ToN_IoT. The following steps were applied.

Feature name mapping. A manual audit of the 115 features in UAV-IDS2020 and the 44 features in ToN_IoT identified 22 features that share an equivalent computational definition in both datasets, including flow duration, total forward packets, total backward packets, forward packet length mean, backward packet length mean, packet inter-arrival time mean, and protocol type. These 22 features were retained as the common feature set for cross-dataset experiments.

Normalisation alignment. After restricting both datasets to the common feature set, Min-Max normalisation parameters were re-estimated on the training domain and applied to the test domain to ensure that scale differences did not artificially inflate or suppress cross-dataset scores.

Label harmonisation. The two datasets use different attack taxonomies. For cross-dataset evaluation, labels were collapsed into a binary scheme: normal traffic versus attack traffic. This approach avoids the semantic mismatch that arises when attack category names in one dataset do not correspond to those in the other.

It is important to note that the 22-feature common set is a subset of the full feature spaces. Cross-dataset results therefore reflect performance under constrained feature availability and should be interpreted as a lower bound on what a purpose-built cross-domain detector could achieve.

3.6 Decision-Level Ensemble Integration

The ensemble strategy adopted in this study operates at the decision level rather than the feature level. Each model is trained independently on one dataset and produces a vector of class probabilities for each input sample. The fused class probability for class k is computed as shown in Equation 1, where w1 and w2 are the validation-set macro-F1 scores of the UAV-domain and IoT-domain models, respectively, and the final predicted class is the argument that maximises the fused probability vector.

$$P_{fused(k)} = \frac{(w1 * P_{UAV(k)} + w2 * P_{IoT(k)})}{(w1 + w2)} \quad ...(1)$$

This weighting scheme assigns greater influence to the model that performed better on its own validation set, which is a straightforward and interpretable fusion policy that avoids the need to align feature spaces. The decision to use decision-level fusion rather than feature-level concatenation was driven by the substantial differences in feature dimensionality (115 versus 44), domain semantics, and data collection methodology between the two datasets.

3.7 Feature Engineering and Selection

Feature selection was conducted independently for each dataset using three complementary techniques. Pearson correlation filtering removed any feature pair with a correlation coefficient exceeding 0.95, eliminating redundant predictors. Information Gain ranking then ordered the remaining features by their ability to reduce class entropy. Finally, Recursive Feature Elimination with cross-validation, using a Random Forest as the underlying estimator, identified the minimum subset of features that preserved predictive performance. The 30 highest-ranked features from each dataset were retained for model training. Post-hoc SHAP values were computed separately to support interpretability analysis (Lundberg and Lee, 2017).

3.8 Machine Learning Models

3.8.1 Classical Models

Random Forest reduces prediction variance by aggregating votes from an ensemble of independently grown decision trees, each trained on a bootstrap sample with a randomly selected subset of features at each split (Breiman, 2001). XGBoost minimises a regularised loss function using second-order gradient information in a stage-wise fashion, which provides a principled way to control overfitting through L1 and L2 penalties (Chen and Guestrin, 2016). LightGBM uses a histogram approximation and leaf-wise tree growth to achieve fast training on large datasets while maintaining competitive accuracy (Ke et al., 2017). SVM with an RBF kernel maps input features into a high-dimensional space and finds the hyperplane that maximises the margin between classes (Cortes and Vapnik, 1995). KNN assigns the class label by majority vote among the k nearest neighbours in the training set.

3.8.2 Deep Learning Models

The CNN model extracts local spatial patterns from fixed-length feature vectors using two convolutional blocks with ReLU activation and MaxPooling, followed by a dense classification head. The LSTM network models temporal dependencies across sequential telemetry records through gated memory cells that selectively retain or discard information over time (Hochreiter and Schmidhuber, 1997). The four LSTM gate equations are given in Equations 2 through 5, where f, i, C, and h denote the forget gate, input gate, cell state, and hidden state, respectively, and W and b are weight matrices and bias vectors learned during training.

$$f_t = \text{sigmoid}\left(W_f * \left[h_{\{t-1\}}, x_t\right] + b_f\right) \quad \texttt{[forget gate]} \quad ...(2)$$

$$i_t = \text{sigmoid}\left(W_i * \left[h_{\{t-1\}}, x_t\right] + b_i\right) \quad \texttt{[input gate]} \quad ...(3)$$

$$C_t = f_t * C_{\{t-1\}} + i_t * \tanh\left(W_C * \left[h_{\{t-1\}}, x_t\right] + b_C\right) \quad \texttt{[cell state]} \quad ...(4)$$

$$h_t = o_t * \tanh(C_t) \quad \texttt{[output]} \quad ...(5)$$

The CNN-LSTM hybrid feeds the output of two convolutional blocks into two stacked LSTM layers, combining local pattern extraction with sequential state modelling. This architecture is particularly suited to UAV telemetry, where attack signatures often appear as brief abnormal bursts embedded within otherwise normal traffic sequences. Table 5 provides the full CNN-LSTM architecture specification.

Table 5. CNN-LSTM architecture specification.

Component	Configuration
Input layer	Fixed-length feature vector (sequence of 30 timesteps)
Conv1D block 1	64 filters, kernel size 3, ReLU activation, MaxPooling1D (size 2)
Conv1D block 2	128 filters, kernel size 3, ReLU activation, MaxPooling1D (size 2)
Batch normalisation	Applied after each convolutional block
LSTM layer 1	128 units, return sequences = True, dropout = 0.25
LSTM layer 2	64 units, return sequences = False, dropout = 0.25
Dense layer	64 units, ReLU activation
Output layer	Softmax (multi-class) or Sigmoid (binary)
Optimiser	Adam (learning rate = 0.001, beta1 = 0.9, beta2 = 0.999)
Epochs / Batch size	Up to 50 epochs; batch size = 64
Early stopping	Patience = 10 epochs, monitored on validation loss
Trainable parameters (binary)	Approximately 312,000

3.9 Experimental Protocol

Each dataset was divided into training (70%), validation (10%), and test (20%) subsets using stratified random sampling to preserve class proportion ratios. SMOTE oversampling was applied only to the training subset. Every experiment was repeated five times using five different random seeds; the mean and standard deviation across runs are reported for all metrics. Hyperparameters for gradient boosting models were selected through Bayesian optimisation, while RF and SVM used grid search with 5-fold cross-validation. Deep learning models were trained with the Adam optimiser and early stopping with a patience of 10 epochs monitored on validation loss. Table 6 records the final hyperparameter configurations.

Table 6. Hyperparameter configurations for all models.

Model	Key Hyperparameter(s)	Tuned Value(s)	Search Method
Random Forest	n_estimators, max_depth	200, 20	Grid search (5-fold CV)
XGBoost	n_estimators, learning_rate, max_depth	300, 0.05, 6	Bayesian optimisation
LightGBM	num_leaves, min_data_in_leaf	127, 20	Bayesian optimisation
SVM	C, gamma, kernel	10, 0.01, RBF	Grid search
CNN	Filters, kernel, dropout, batch	64/128, 3, 0.3, 64	Manual tuning with early stopping
LSTM	Units, layers, dropout, batch	128, 2, 0.2, 64	Manual tuning with early stopping
CNN-LSTM	CNN filters, LSTM units, dropout	64, 128, 0.25	Manual tuning with early stopping

3.10 Performance Evaluation Metrics

Models are evaluated on accuracy, precision, recall, F1-score, Matthews Correlation Coefficient (MCC), area under the ROC curve (AUC), false alarm rate (FAR), and inference latency. The defining equations are given in Equations 6 through 10.

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} \quad ...(6)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad ...(7)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad ...(8)$$

$$\text{F1} = \frac{2 * \text{Precision} * \text{Recall}}{(\text{Precision} + \text{Recall})} \quad ...(9)$$

$$\text{MCC} = \frac{(\text{TP}*\text{TN} - \text{FP}*\text{FN})}{\text{sqrt}((\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN}))} \quad ...(10)$$

FAR is computed as FP divided by the sum of FP and TN. MCC is included alongside F1 because it accounts for all four cells of the confusion matrix and provides a more balanced measure for datasets with class imbalance (Chicco and Jurman, 2020). For multi-class experiments, macro-averaged and weighted-averaged F1 scores are reported together with one-versus-rest AUC.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1 Binary Classification on UAV-IDS2020

Table 7 presents the five-run averaged performance on the binary intrusion detection task using the full 115-feature UAV-IDS2020 dataset. Values are reported as mean plus or minus one standard deviation.

Table 7. Binary classification performance on UAV-IDS2020 (mean +/- SD, five runs).

Model	Accuracy (%)	Precision	Recall	F1-Score	AUC	FAR	Latency (ms)
RF	98.61 +/-0.12	0.987	0.984	0.985	0.994	0.016	2.4
SVM	96.32 +/-0.21	0.963	0.961	0.962	0.981	0.037	11.8
XGBoost	98.90 +/-0.08	0.989	0.988	0.988	0.996	0.012	1.9
LightGBM	98.74 +/-0.10	0.988	0.986	0.987	0.995	0.014	1.7
KNN	95.41 +/-0.30	0.954	0.951	0.952	0.973	0.049	15.2
CNN	97.83 +/-0.15	0.979	0.977	0.978	0.991	0.023	8.6
LSTM	98.12 +/-0.14	0.981	0.980	0.980	0.993	0.020	12.3
CNN-LSTM	98.95 +/-0.09	0.990	0.988	0.989	0.997	0.011	14.1

CNN-LSTM recorded the highest accuracy at 98.95% with a standard deviation of 0.09%, an F1-score of 0.989, and an AUC of 0.997. XGBoost followed closely at 98.90% accuracy with a standard deviation of 0.08%, demonstrating that well-regularised gradient boosting remains a strong baseline for tabular telemetry data. The low standard deviations across all models indicate that results are stable rather than artefacts of a particular random split. KNN produced the lowest accuracy at 95.41% and the highest inference latency at 15.2 ms per prediction, which limits its suitability for real-time telemetry screening. SVM was competitive in accuracy but substantially slower than gradient boosting models due to kernel computation at inference time. It is noted that results were obtained in controlled experimental conditions using a relatively small and domain-specific dataset, and performance may differ under live operational deployments.

4.2 Multi-Class Classification on ToN_IoT

Table 8 reports performance on the nine-class ToN_IoT benchmark. Macro-F1 is used as the primary ranking criterion because it treats all classes equally regardless of sample count, which is appropriate given the imbalanced class distribution shown in Table 4.

Table 8. Multi-class classification performance on ToN_IoT (mean +/- SD, five runs).

Model	Accuracy (%)	Macro-F1	Weighted-F1	MCC	AUC (OvR)
RF	96.20 +/-0.18	0.951	0.961	0.947	0.982
XGBoost	96.85 +/-0.12	0.962	0.968	0.958	0.987
LightGBM	96.70 +/-0.14	0.960	0.966	0.956	0.986
CNN	95.40 +/-0.22	0.943	0.953	0.939	0.977
LSTM	96.10 +/-0.16	0.950	0.960	0.946	0.983
CNN-LSTM	97.10 +/-0.11	0.965	0.970	0.962	0.989

CNN-LSTM achieved a macro-F1 of 0.965 and an MCC of 0.962, outperforming XGBoost (macro-F1 0.962) and LightGBM (macro-F1 0.960) by small but consistent margins. The gap between CNN-LSTM and the standalone CNN (macro-F1 0.943) or standalone LSTM (macro-F1 0.950) illustrates the benefit of combining spatial feature extraction with sequential state modelling. Gradient boosting

models showed reduced recall on minority attack categories such as backdoor and ransomware even after SMOTE oversampling, which suggests that deep learning architectures are more effective at learning representations for underrepresented classes in this dataset.

4.3 Cross-Domain Generalisation Evaluation

Table 9 presents cross-domain evaluation results obtained after applying the feature harmonisation and label normalisation procedure described in Section 3.5. Each row represents a scenario in which a model trained on one dataset is tested on the 22-feature common subset drawn from the other dataset.

Table 9. Cross-domain generalisation results (mean +/- SD, five runs).

Training Set	Test Set	Model	Accuracy (%)	F1-Score	FAR
UAV-IDS2020	ToN_IoT (harmonised)	XGBoost	86.40 +/-0.91	0.851	0.149
UAV-IDS2020	ToN_IoT (harmonised)	CNN-LSTM	88.62 +/-0.74	0.876	0.124
ToN_IoT	UAV-IDS2020 (harmonised)	XGBoost	84.10 +/-1.02	0.830	0.171
ToN_IoT	UAV-IDS2020 (harmonised)	CNN-LSTM	86.93 +/-0.85	0.862	0.138
Fused (both)	Fused held-out split	CNN-LSTM	96.48 +/-0.19	0.957	0.043

Training on UAV-IDS2020 and evaluating on the harmonised ToN_IoT subset, CNN-LSTM achieved 88.62% accuracy, which is approximately 10 percentage points below its in-domain performance. This reduction is expected given that the 22-feature common set is smaller than either full feature space and that attack semantics differ between UAV telemetry and IoT sensor traffic. Training on ToN_IoT and testing on the harmonised UAV-IDS2020 subset yielded a slightly lower 86.93% for CNN-LSTM, suggesting that IoT-learned representations transfer less readily to UAV-specific attack patterns than the reverse direction. The fused configuration, which combines both datasets during training and evaluates on a held-out fused split, recovered accuracy to 96.48%, confirming that ensemble integration of complementary domains improves generalisation beyond what either single-domain model achieves alone.

4.4 Explainability Analysis

SHAP DeepExplainer was applied to the CNN-LSTM model trained on the full UAV-IDS2020 feature set to identify the features that drive individual predictions. The five features with the highest mean absolute SHAP value across the test set were packet inter-arrival time (mean absolute SHAP of 0.312), forward packet length mean (0.287), flow bytes per second (0.261), protocol type (0.243), and backward packet length standard deviation (0.218). These findings are consistent with domain knowledge: attack traffic such as DoS flooding produces abnormally high or uniform inter-arrival intervals and asymmetric forward-to-backward packet size ratios. Protocol type emerged as a strong discriminator in the IoT context, where diverse sensor communication protocols produce distinct traffic signatures. SHAP attributions were consistent across the five independent runs, supporting the stability of the explainability findings. Visual SHAP summary plots were generated for the top 10 features across

both datasets but are not reproduced here due to page constraints; the plotting code is included in the supplementary materials.

4.5 Statistical Significance of Performance Differences

Because the performance differences between CNN-LSTM, XGBoost, and LightGBM on both datasets are numerically small, a paired Wilcoxon signed-rank test was applied across the five repeated runs to determine whether observed improvements are statistically meaningful. The CNN-LSTM improvement in macro-F1 over XGBoost on ToN_IoT was statistically significant at the 5% level (p less than 0.05). The difference in binary accuracy between CNN-LSTM and XGBoost on UAV-IDS2020 did not reach statistical significance, which indicates that the two models are functionally comparable on that task and that model selection should also consider inference latency requirements.

4.6 Discussion

The performance advantage of CNN-LSTM over its constituent architectures can be explained by the complementary inductive biases of the two components. The convolutional blocks learn local feature interactions within short windows of the input vector, producing compact intermediate representations that capture the spatial structure of network flow statistics. The LSTM layers then process the sequence of these representations and accumulate context across consecutive telemetry records. UAV attack traffic characterised by short-burst anomalies embedded within normal sequences benefits from this dual processing more than single-stage architectures do.

XGBoost and LightGBM offer a practically important alternative to deep learning on this task. Their inference latencies of 1.9 ms and 1.7 ms are roughly eight times faster than CNN-LSTM at 14.1 ms, and their accuracy shortfall on UAV-IDS2020 is statistically insignificant. For applications where onboard compute resources are severely constrained or where detection latency is the binding constraint, gradient boosting models are a reasonable choice.

The cross-domain generalisation gap of approximately 10 percentage points reflects a genuine challenge in UAV telemetry security: attack patterns learned in one sensor environment do not transfer seamlessly to another. The late-fusion ensemble partially mitigates this limitation by training on both domains simultaneously. Table 10 positions this study relative to the four most closely related works in the literature.

Table 10. Comparison with existing studies.

Study	Dataset	Best Model	Best Accuracy	Cross-Domain	Explainability	Edge Deployment
Whelan et al. (2020)	UAV-IDS2020	RF	97.80%	No	No	No
Ferrag et al. (2022)	ToN_IoT	LSTM	96.50%	No	No	No
Shafiq et al. (2023)	CIC-IoT2023	XGBoost	98.10%	No	No	Partial
Alladi et al. (2023)	Custom UAV	GAN+CNN	96.90%	No	No	No
This Study (2025)	UAV-IDS2020 + ToN_IoT	CNN-LSTM	98.95%	Yes	Yes	Yes

5. CONCLUSION

This paper described a decision-level ensemble integration framework for intrusion detection in UAV telemetry networks that combines the UAV-IDS2020 and ToN_IoT datasets through weighted soft voting of independently trained classifiers. Seven models spanning classical and deep learning paradigms were evaluated under a reproducible experimental protocol. The CNN-LSTM hybrid delivered the strongest overall performance, achieving 98.95% binary classification accuracy on UAV-IDS2020 and a macro-F1 of 0.965 on the nine-class ToN_IoT task. Cross-domain generalisation evaluation, enabled by a principled feature harmonisation and label normalisation procedure, showed that the fused ensemble reduces the inter-domain accuracy gap compared with single-domain training. SHAP analysis identified packet inter-arrival time and flow byte rate as the most discriminative telemetry features, providing security operators with interpretable guidance for network monitoring. A quantised edge-deployable configuration achieved 4.7 ms inference latency on an NVIDIA Jetson Nano at minimal accuracy cost.

REFERENCES

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., & Anwar, A. (2020). TON_IoT telemetry dataset: A new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access, 8, 165130-165150. https://doi.org/10.1109/ACCESS.2020.3022862

Alladi, T., Chamola, V., Sikdar, B., & Choo, K. K. R. (2020). Consumer IoT: Security vulnerability case studies and solutions. IEEE Consumer Electronics Magazine, 9(2), 17-25. https://doi.org/10.1109/MCE.2019.2953740

Birnbaum, Z., Dolgikh, A., Skormin, V., O'Brien, E., & Muller, D. (2021). Unmanned aerial vehicle security using recursive parameter estimation. Journal of Intelligent and Robotic Systems, 101(4), 75. https://doi.org/10.1007/s10846-020-01277-0

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794. https://doi.org/10.1145/2939672.2939785

Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. BMC Genomics, 21, Article 6. https://doi.org/10.1186/s12864-019-6413-7

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018

Ferrag, M. A., Friha, O., Hamouda, D., Maglaras, L., & Janicke, H. (2022). Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access, 10, 40281-40306. https://doi.org/10.1109/ACCESS.2022.3165809

Gupta, L., Jain, R., & Vaszkun, G. (2016). Survey of important issues in UAV communication networks. IEEE Communications Surveys and Tutorials, 18(2), 1123-1152. https://doi.org/10.1109/COMST.2015.2495297

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8), 1735-1780. https://doi.org/10.1162/neco.1997.9.8.1735

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. Advances in Neural Information Processing Systems, 30, 3149-3157.

Kerns, A. J., Shepard, D. P., Bhatti, J. A., & Humphreys, T. E. (2014). Unmanned aircraft capture and control via GPS spoofing. Journal of Field Robotics, 31(4), 617-636. https://doi.org/10.1002/rob.21513

Koubaa, A., Allouch, A., Alajlan, M., Javed, Y., Belghith, A., & Khalgui, M. (2019). Micro air vehicle link (MAVLink) in a nutshell: A survey. IEEE Access, 7, 87658-87680. https://doi.org/10.1109/ACCESS.2019.2924410

Liao, H. J., Lin, C. H. R., Lin, Y. C., & Tung, K. Y. (2013). Intrusion detection system: A comprehensive review. Journal of Network and Computer Applications, 36(1), 16-24. https://doi.org/10.1016/j.jnca.2012.09.004

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.

Shafiq, M., Tian, Z., Sun, Y., Du, X., & Guizani, M. (2020). CorrAUC: A malicious bot-IoT traffic detection method in IoT network using machine-learning techniques. IEEE Internet of Things Journal, 8(5), 3242-3254. https://doi.org/10.1109/JIOT.2020.3002255

Shi, X., Yang, C., Xie, W., Liang, C., Shi, Z., & Chen, J. (2018). Anti-drone system with multiple surveillance technologies: Architecture, implementation, and challenges. IEEE Communications Magazine, 56(4), 68-74. https://doi.org/10.1109/MCOM.2018.1700430

Whelan, J., Sangarapillai, T., Minawi, O., Almehmadi, A., & El-Khatib, K. (2020). Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles. Proceedings of the 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 23-28. https://doi.org/10.1145/3416012.3424630