Robust Hybrid Stack for RF-Based Drone Controller Classification with Feature-Degradation-Tolerant Fusion

Luis Paulo Albuquerque Guedes[1], Pedro H. M. Guedes[2] e Richard Barbosa Nobre[3]
[1]Comando do 7º Distrito Naval (Com7DN), Brasília/DF - Brasil
[2]Instituto de Pesquisas da Marinha (IPqM), Rio de Janeiro/RJ - Brasil
[3]Centro de Guerra Acústica e Eletrônica da Marinha (CGAEM), Rio de Janeiro/RJ - Brasil

***Abstract*— This work proposes a robust hybrid model for drone controller classification based on radio-frequency (RF) features. The approach combines high-performance classical classifiers, perturbation-tolerant models, and a logistic meta-classifier within a supervised *stacking* architecture. The goal is to preserve high accuracy under clean conditions while improving stability under controlled degradation of the extracted feature representation. Experiments conducted on the *MPACT DroneRC RF Dataset* show that HistGradientBoosting achieves the best performance on unperturbed data, with 0.9976 accuracy and 0.9976 Macro-F1. Under the strongest evaluated feature-space perturbation, its Macro-F1 drops to 0.4849. In contrast, the proposed *Robust Hybrid Stack* maintains 0.9965 Macro-F1 on clean data and achieves the best performance under the highest perturbation level, preserving 0.9525 Macro-F1. These results indicate that hybrid feature-level fusion is a promising alternative when the extracted RF descriptors are subject to controlled feature degradation or measurement uncertainty. The reported robustness refers specifically to perturbations applied after feature extraction and does not constitute a direct evaluation of additive noise or interference applied to the original RF waveform.**

***Keywords*— Drones, RF classification, feature fusion, hybrid model, feature degradation, machine learning.**

I. Introduction

Drone detection and classification are important for surveillance, defense, and critical-infrastructure protection. Radio-frequency (RF) sensing enables passive monitoring without requiring visual contact or favorable illumination. RF classification relies on spectral, temporal, bandpower, and transient features, which may be affected by noise, interference, and environmental variability.

This work investigates multiclass controller classification using the *MPACT DroneRC RF Dataset*. Classical models, tabular neural networks, a tabular Transformer, and the proposed *Robust Hybrid Stack* are compared. The proposed model combines HistGradientBoosting, Random Forest, Extra Trees, Logistic Regression, and KNN through a logistic meta-classifier using class probabilities, confidence, margin, and entropy, with feature-space perturbations included during training.

The hypothesis is that complementary models can balance clean-data performance, computational efficiency, and tolerance to feature degradation. HistGradientBoosting achieves the best clean-data performance, whereas the *Robust Hybrid Stack* obtains the highest Macro-F1 under the strongest evaluated feature-space perturbation. The main contributions are the proposed hybrid classifier, a unified comparison of different model families, and a joint analysis of predictive performance, calibration, operational cost, feature-space robustness, and feature importance.

L. Guedes, luis.guedes@coppe.ufrj.br; P. Guedes, pedrohmguedes@gmail.com; R. Nobre, richard.b.nobre@gmail.com.

II. Related Work

RF fingerprinting identifies wireless emitters through hardware-dependent imperfections and signal characteristics. Jagannath et al. [1] review feature-based and deep-learning methods, datasets, and applications in signal intelligence, device authentication, and wireless security.

For UAV classification, Al-Sa'd et al. [2] investigated deep-learning methods, while Allahham et al. [3] introduced the DroneRF dataset. Ezuma et al. [4], [5] studied micro-UAV RF fingerprints under Wi-Fi and Bluetooth interference.

Several approaches use time-frequency representations. Fu and He [6] applied Gramian Angular Fields and CNNs, Nguyen et al. [7] combined CNNs with CFAR filtering, and Kirakosyan et al. [8] addressed real-time spectrogram classification. Guedes et al. [9] compared VGG-16 and ViT-Tiny on STFT-based DroneDetect spectrograms.

Unlike spectrogram-based approaches, this work focuses on numerical RF descriptors. It compares classical models, tree ensembles, deep tabular networks, and a tabular Transformer. The evaluation includes Random Forests [10], Extra Trees [11], Gradient Boosting [12], deep tabular models [13], residual learning [14], and Transformer-based attention [15].

The proposed *Robust Hybrid Stack* follows stacked generalization [16], combining ensemble, linear, and instance-based models such as KNN [17]. Their probabilities and uncertainty-related descriptors are used as meta-features by a logistic meta-classifier trained to improve tolerance to feature degradation.

The evaluation also considers calibration through ECE, NLL, and Brier score [18], bootstrap confidence intervals [19], and controlled feature-space perturbations motivated by noise-injection regularization [20]. Thus, the study provides a feature-level benchmark and a hybrid stacking approach for RF drone controller classification under degraded feature conditions.

III. Materials and Methods

The experimental methodology was implemented in Python for training, evaluating, and comparing classical machine-learning models, deep networks for tabular data, a Transformer-based model, and the proposed robust hybrid model based on *stacking*. The pipeline operated on previously extracted radio-frequency features from the *MPACT DroneRC RF Dataset*, focusing on multiclass classification of the controller identifier, defined by the `controller_id` variable.

A. Dataset and Classification Task

The dataset was loaded from a CSV file containing RF samples and numerical descriptors. Only valid records were retained; when available, the `read_status` column was used to select entries marked as `ok`, and duplicate records were removed based on `full_path`. In the reported experiment, the final dataset contained 3,400 valid samples distributed across 17 classes, with 78 total columns and 65 numerical features used for modeling.

B. Feature Selection and Preprocessing

Feature selection removed metadata and non-informative fields, such as file paths, filenames, manufacturer, model, signal index, reading metadata, error messages, and the target variable. The remaining attributes were converted to numerical format, with median imputation for missing values and zero-mean, unit-variance standardization for scale-sensitive models. Preprocessing parameters were fitted only on the training set and subsequently applied to the validation and test sets.

C. Data Partitioning

The data were split into training, validation, and test sets using stratified sampling, preserving the class distribution. The test set corresponded to 25% of the samples, the validation set to 15% of the full dataset, and the remaining samples were used for training. To reduce class-related bias, the maximum number of samples per class was limited to 200. All splits used a fixed random seed equal to 42.

The adopted partitioning guarantees disjoint subsets at the sample level; however, it does not explicitly enforce group-wise separation according to acquisition session, recording origin, or other acquisition-level identifiers. Therefore, if multiple feature vectors originate from related acquisition conditions, correlated samples may potentially occur in different subsets. This limitation should be considered when interpreting the reported generalization performance. Group- and session-aware partitioning will be investigated in future work to provide a stricter assessment under acquisition-independent conditions.

D. Evaluated Models

Four model families were evaluated: classical methods, deep neural networks for tabular data, a tabular Transformer, and the proposed robust hybrid model. The classical models included *Dummy Prior*, Logistic Regression, linear SVM, RBF-kernel SVM, KNN, Random Forest, Extra Trees, Histogram-based Gradient Boosting, and Scikit-learn MLP. The deep models implemented in PyTorch included *Deep MLP*, *Tabular ResNet*, and *Feature Transformer*.

In addition to these individual models, a robust hybrid classifier named *Robust Hybrid Stack* was proposed. This model combines five complementary specialists: HistGradientBoosting, Random Forest, Extra Trees, Logistic Regression, and KNN. The selection of these specialists was guided by preliminary results: HistGradientBoosting achieved the best performance under clean conditions; Random Forest and Extra Trees showed high and stable performance; Logistic Regression provided low latency; and KNN exhibited greater robustness to perturbations in the feature space. The goal of the hybrid model is to combine these properties into a single robustness-oriented classifier.

E. Proposed Robust Hybrid Model

The proposed hybrid model was implemented through supervised *stacking*. Initially, the five base specialists were trained on internal partitions of the training set using five-fold stratified cross-validation. To avoid information leakage, the inputs to the meta-classifier were generated from out-of-fold predictions. Each specialist produced multiclass probabilities for the 17 classes of the problem.

From the probabilities produced by each specialist, meta-features were constructed using the class probabilities, maximum confidence, margin between the two largest probabilities, and predictive entropy. Since five specialists and 17 classes were used, the final meta-feature vector contained 100 attributes. The adopted meta-classifier was Logistic Regression with class balancing.

To improve the robustness of the hybrid model, meta-features were also generated from perturbed versions of the internal validation attributes. The perturbations were applied in the feature space using noise levels of 0.01 and 0.03, scaled by the standard deviation of each attribute in the training set. Thus, the meta-classifier was trained with both clean and perturbed predictions, resulting in 7,650 meta-training examples. After training the meta-classifier, all specialists were refitted using the combined training and validation set, composed of 2,550 samples. The final evaluation was performed on the test set, containing 850 samples.

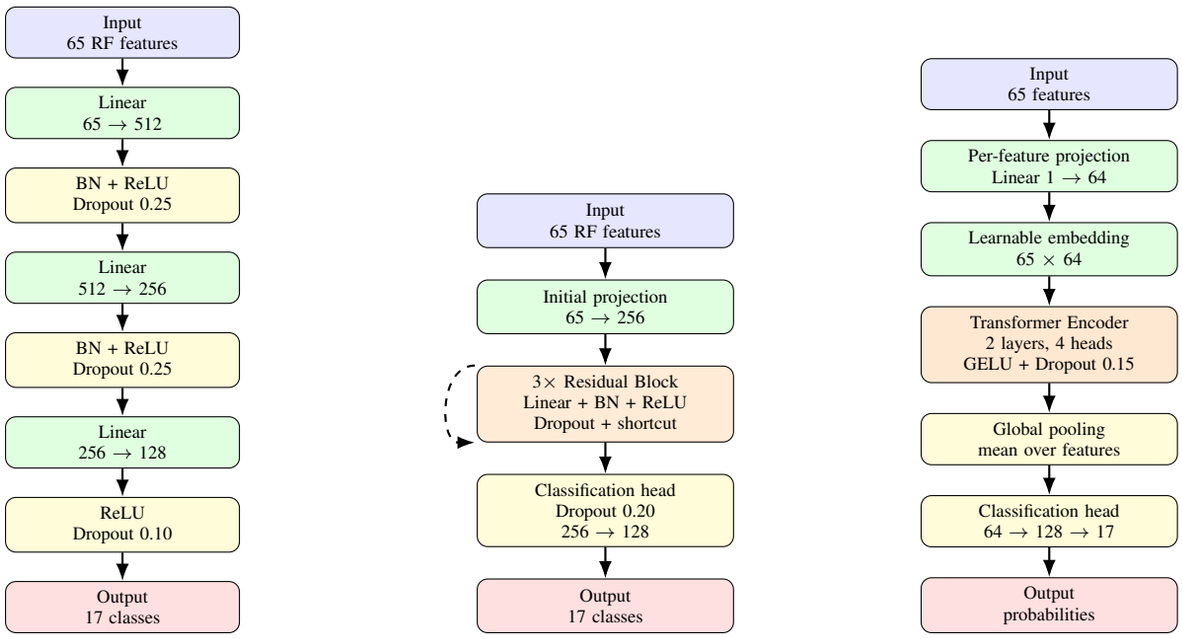
Fig. 2 presents the conceptual architecture of the proposed hybrid model.

F. Training Protocol

The classical models were trained using Scikit-learn implementations. Median imputation was applied to Logistic Regression, SVMs, and neighborhood-based models. For scale-sensitive models, feature standardization was also employed. In addition, some classifiers used class-balancing mechanisms, such as `class_weight="balanced"`, to reduce potential effects of class imbalance during training.

The deep models and the *Transformer* were implemented in PyTorch and trained using the AdamW optimizer, a learning rate of 3×10^{-4}, weight decay of 1×10^{-4}, batch size of 64, and a maximum limit of 60 training epochs.

The adopted loss function was categorical cross-entropy. To select the best training point, early stopping with a patience of 10 epochs was used, monitoring validation performance through the Macro-F1 metric. At the end of training, the best model state was restored before final evaluation on the test set.



(a) Deep MLP **(b) Tabular ResNet** **(c) Feature Transformer**

Figure 1. Deep architectures evaluated for RF feature classification: (a) *Deep MLP*, based on dense layers; (b) *Tabular ResNet*, with fully connected residual blocks; and (c) *Feature Transformer*, which represents tabular attributes as tokens processed by attention.

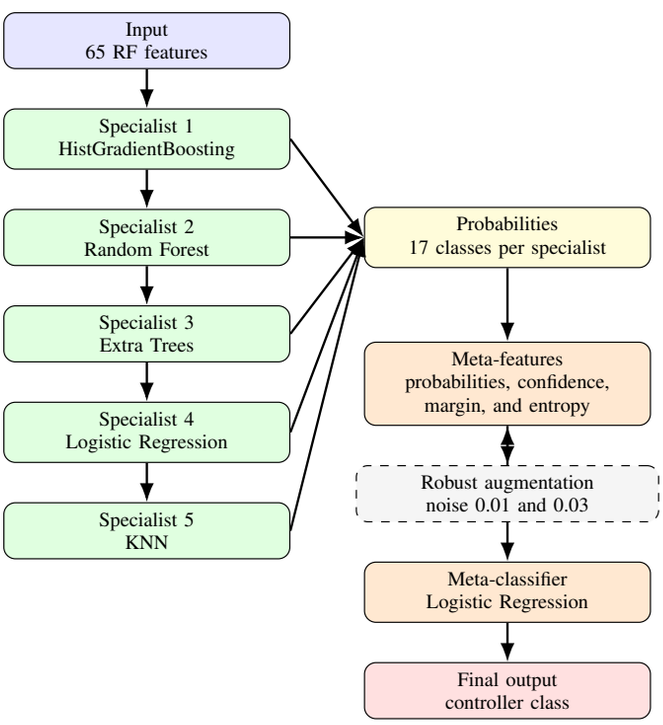


Figure 2. Conceptual architecture of the *Robust Hybrid Stack* model. The model combines five complementary specialists and uses their probabilities, confidence, margin, and entropy as meta-features for a logistic meta-classifier trained with robust noise augmentation in the feature space.

For the *Robust Hybrid Stack*, training was performed in two stages. In the first stage, out-of-fold predictions from the base specialists were generated using five-fold stratified cross-validation. These predictions were used to construct the meta-training set. In the second stage, a logistic meta-classifier was fitted on the generated meta-features. To improve the robustness of the final classifier, meta-training included perturbed versions of the attributes, with noise levels of 0.01 and 0.03. After this process, the specialists were refitted using the complete training and validation set.

G. Evaluation Metrics

The evaluation included predictive metrics, calibration metrics, operational measures, and robustness tests. For classification performance, accuracy, balanced accuracy, Macro-F1, weighted F1, macro precision, macro recall, top-2 accuracy, top-3 accuracy, and multiclass ROC-AUC under the *one-vs-rest* scheme were considered. Since the task involves 17 classes, Macro-F1 was adopted as a central metric because it assigns equal weight to each class, regardless of its frequency in the test set.

Probability calibration was assessed using Expected Calibration Error (ECE), Negative Log-Likelihood (NLL), and the Brier score. These metrics make it possible to verify whether the confidence assigned by the model to its predictions is aligned with the empirical frequency of correctness, which is relevant for operational applications in which decision uncertainty must be interpretable.

In addition to predictive metrics, practical deployment-related quantities were evaluated, including training time, inference time, latency per sample, throughput in samples per second, model size, and confusion cost. The confusion cost was defined as

$$C_{\text{conf}} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}(y_i \neq \hat{y}_i), \tag{1}$$

where y_i is the true label, $\hat{y}_i$ is the predicted label, N is the total number of evaluated samples, and $\mathbb{I}(\cdot)$ is the indicator function.

To rank the models while jointly considering predictive performance and deployment feasibility, the following operational score was computed:

$$S_{\text{op}} = 0.50\,F1_{\text{macro}} + 0.20\,Acc_{\text{bal}} + 0.10\,Acc_{\text{top3}} + 0.10\,S_{\text{lat}} + 0.10\,S_{\text{size}}. \tag{2}$$

In this score, $F1_{\text{macro}}$ denotes Macro-F1, Acc_{bal} denotes balanced accuracy, Acc_{top3} denotes top-3 accuracy, S_{lat} is the latency term, and S_{size} is the model-size term. The latter two were normalized inversely proportional to latency and model size, so that faster and more compact models received higher operational scores:

$$S_{\text{lat}} = \frac{1}{1 + L/10}, \qquad S_{\text{size}} = \frac{1}{1 + M/50}, \tag{3}$$

where L is the latency in milliseconds per sample and M is the model size in megabytes.

Finally, robustness to feature degradation was evaluated through Gaussian perturbations applied directly to the extracted feature space. The attributes of the test set were perturbed at levels equal to 0, 0.01, 0.03, 0.05, 0.10, and 0.20, defined as fractions of the standard deviation of each feature in the training set. This analysis allowed the stability of the models to be compared under controlled degradation of the numerical RF descriptors. Accordingly, the term robustness in this study refers to tolerance to feature-space perturbations and should not be interpreted as robustness to additive noise, interference, channel effects, or other impairments applied directly to the original RF waveform.

H. Experimental Outputs

For each model, the pipeline stores the main artifacts required for analysis and reproducibility, including the trained model, classification report, confusion matrix, predictions, PyTorch training history, consolidated metric tables, and comparative plots. These outputs support the joint assessment of predictive performance, calibration, latency, throughput, model size, and operational score.

I. Methodological Summary

This study compares classical models, deep tabular networks, a tabular Transformer, and the proposed *Robust Hybrid Stack* under a unified protocol for preprocessing, stratified splitting, supervised training, and multicriteria evaluation. The protocol assesses both RF classification performance and deployment feasibility in terms of latency, model size, calibration, and robustness to feature-space perturbations.

The *Robust Hybrid Stack* combines specialists with complementary properties, including high clean-data accuracy, statistical stability, low latency, and tolerance to feature-space perturbations. Thus, the experiments compare three practical regimes: maximum performance on clean data, best operational trade-off, and robustness under controlled feature degradation.

IV. Results and Discussion

This section reports the results for drone controller classification using RF features from the MPACT DroneRC dataset. Classical models, tabular neural networks, a tabular Transformer, and the proposed *Robust Hybrid Stack* were evaluated in terms of classification performance, calibration, operational efficiency, statistical stability, robustness to feature-space perturbations, and feature importance.

Table I summarizes the main results. Under clean conditions, HistGradientBoosting achieved the best performance, with accuracy and Macro-F1 of 0.9976. Random Forest and Extra Trees also performed strongly, with Macro-F1 scores of 0.9965 and 0.9929, respectively. The proposed *Robust Hybrid Stack*, which combines HistGradientBoosting, Random Forest, Extra Trees, Logistic Regression, and KNN through a logistic meta-classifier, reached a Macro-F1 of 0.9965, closely matching the best ensembles while improving robustness under severe feature perturbations. Tabular ResNet obtained the highest operational score, combining a Macro-F1 of 0.9811, latency of 0.0786 ms per sample, and model size of 1.75 MB.

Table II summarizes the best model for each evaluation criterion. This analysis shows that no single model dominates across all aspects. HistGradientBoosting is superior under clean conditions and achieves the best calibration; Tabular ResNet provides the best operational trade-off; Logistic Regression is the lightest and fastest useful alternative; and the *Robust Hybrid Stack* stands out under the most degraded feature condition, achieving the highest Macro-F1 under the strongest evaluated feature-space perturbation.

This result is central to interpreting the contribution of the proposed model. Although the *Robust Hybrid Stack* does not outperform HistGradientBoosting in clean accuracy or Tabular ResNet in operational score, it achieves the best absolute performance under the highest feature-space perturbation level. Thus, its main contribution is not maximizing performance under ideal conditions, but preserving discriminative capability when the extracted RF feature representation is degraded.

The bootstrap analysis in Table III confirms the stability of the best-performing models. HistGradientBoosting achieved a Macro-F1 of 0.9976, with a 95% confidence interval between 0.9939 and 1.0000. Random Forest showed a similar interval, indicating statistically close performance to the best model. The *Robust Hybrid Stack* also presented a narrow interval, with mean Macro-F1 of 0.9964 and a 95% interval between 0.9919 and 1.0000, reinforcing its stability on the test set.

Probability calibration also favored HistGradientBoosting, as shown in Table IV. The model achieved ECE of 0.0017, NLL of 0.0105, and Brier score of 0.0037, indicating that its predicted probabilities are well aligned with empirical correctness. The *Robust Hybrid Stack* also showed strong calibration, with ECE of 0.0062, NLL of 0.0125, and Brier score of 0.0064. These values indicate that the hybrid model preserves reliable probabilistic behavior, although at higher operational cost.

TABLE IV
CALIBRATION OF THE MAIN MODELS.

Model	Macro-F1	ECE	NLL	Brier
HistGradientBoosting	0.9976	0.0017	0.0105	0.0037
Robust Hybrid Stack	0.9965	0.0062	0.0125	0.0064
Tabular ResNet	0.9811	0.0090	0.0892	0.0295
Deep MLP	0.9754	0.0139	0.1509	0.0404
Extra Trees	0.9929	0.0239	0.0371	0.0139
Random Forest	0.9965	0.0278	0.0374	0.0126
Logistic Regression	0.9753	0.0697	0.2256	0.0645
Linear SVM	0.9741	0.1617	0.2794	0.0812

Table V shows the robustness analysis under Gaussian

TABLE I

COMPACT COMPARISON OF THE MAIN EVALUATED MODELS.

Model	Family	Acc.	Macro-F1	Top-3 Acc.	ECE	Lat. (ms)	Size (MB)	Op. Score
HistGradientBoosting	Classical	0.9976	0.9976	1.0000	0.0017	0.8425	6.77	0.9787
Random Forest	Classical	0.9965	0.9965	1.0000	0.0278	0.2895	9.83	0.9783
Robust Hybrid Stack	Proposed hybrid	0.9965	0.9965	1.0000	0.0062	2.0285	57.20	0.9273
Extra Trees	Classical	0.9929	0.9929	1.0000	0.0239	0.3706	34.46	0.9507
Tabular ResNet	Deep learning	0.9812	0.9811	0.9988	0.0090	0.0786	1.75	0.9825
Logistic Regression	Classical	0.9753	0.9753	0.9976	0.0697	0.0079	0.01	0.9824
Feature Transformer	Transformer	0.9518	0.9505	0.9953	0.0242	0.3628	0.45	0.9607

TABLE II

BEST MODEL BY EVALUATION CRITERION.

Criterion	Best model	Interpretation	Value	Observation
Highest clean accuracy	HistGradientBoosting	Best classifier on unperturbed data	0.9976	Best absolute performance
Highest clean Macro-F1	HistGradientBoosting	Best balance across the 17 classes	0.9976	Superior to neural models
Best calibration	HistGradientBoosting	Lowest expected calibration error	ECE = 0.0017	Most reliable probabilities
Best operational score	Tabular ResNet	Best trade-off between performance, latency, and size	0.9825	Most balanced model for deployment
Lowest useful latency	Logistic Regression	Fastest inference with strong performance	0.0079 ms	Suitable for computational constraints
Smallest useful model	Logistic Regression	Most compact model with high performance	0.01 MB	Lightweight embedded alternative
Lowest noise-induced drop	KNN	Smallest degradation between noise 0 and 0.20	0.0163	Stable, but less accurate on clean data
Highest Macro-F1 under severe noise	Robust Hybrid Stack	Best performance in the worst perturbation scenario	0.9525	Main advantage of the proposed model

TABLE III

BOOTSTRAP 95% CONFIDENCE INTERVALS.

Model	Acc. 95% CI	Bal. Acc. 95% CI	Macro-F1 95% CI	Weighted-F1 95% CI
HistGradientBoosting	0.9977 [0.9941, 1.0000]	0.9977 [0.9940, 1.0000]	0.9976 [0.9939, 1.0000]	0.9977 [0.9941, 1.0000]
Random Forest	0.9965 [0.9918, 1.0000]	0.9965 [0.9917, 1.0000]	0.9964 [0.9917, 1.0000]	0.9965 [0.9918, 1.0000]
Robust Hybrid Stack	0.9964 [0.9918, 1.0000]	0.9964 [0.9919, 1.0000]	0.9964 [0.9919, 1.0000]	0.9964 [0.9917, 1.0000]
Extra Trees	0.9930 [0.9871, 0.9976]	0.9930 [0.9869, 0.9979]	0.9929 [0.9868, 0.9978]	0.9930 [0.9870, 0.9976]
Tabular ResNet	0.9811 [0.9717, 0.9894]	0.9810 [0.9715, 0.9893]	0.9808 [0.9710, 0.9891]	0.9810 [0.9709, 0.9894]
Logistic Regression	0.9753 [0.9635, 0.9859]	0.9753 [0.9644, 0.9853]	0.9751 [0.9640, 0.9850]	0.9753 [0.9639, 0.9858]

feature perturbations. HistGradientBoosting achieved the best clean performance, but its Macro-F1 dropped from 0.9976 to 0.4849 at the highest noise level. In contrast, the *Robust Hybrid Stack* retained a Macro-F1 of 0.9525, outperforming all individual models. However, without a conventional stacking baseline trained under the same conditions, the specific contribution of perturbation-based meta-training cannot be isolated.

The robustness analysis shows that KNN had the smallest performance drop under perturbation, although with lower clean-data performance. In contrast, the *Robust Hybrid Stack* combined near-maximum clean performance with the highest Macro-F1 under the strongest feature-space degradation, making it the best evaluated option when tolerance to controlled feature perturbations is prioritized.

These results are limited to perturbations applied after RF feature extraction and therefore do not demonstrate robustness to physical noise, interference, propagation effects, or receiver impairments in the original RF signal. Moreover, a direct ablation between conventional stacking and perturbation-aware meta-training was not performed, so the specific contribution of the robust training strategy cannot be isolated from the effect of the ensemble itself.

The feature-group ablation in Table VI shows that the x group was the most informative in isolation, reaching a Macro-F1 of 0.9894 with 22 features, followed by *bandpower* with 0.9812. Removing the x group produced the largest performance reduction, although overall performance remained high.

Finally, Table VII presents the most relevant features according to permutation importance. The most important variables were `bandpower_1_1p5ghz`, `transient_start_idx`, `bandpower_4_5ghz`, `x_kurtosis`, and `bandpower_0_0p5ghz`. These results show that controller discrimination depends mainly on spectral energy distribution and temporal statistics of the signal.

In summary, the results show that RF-based controller classification is highly feasible with the extracted features. HistGradientBoosting is the best option when the objective is to maximize performance and calibration under clean conditions. Tabular ResNet provides the best operational trade-off due to its low latency and compact size. The *Robust Hybrid Stack*, in turn, represents the main robustness-oriented contribution: although it is not the best model across all criteria, it achieved the highest Macro-F1 under severe feature-space perturbation, maintaining 0.9525 at the highest evaluated level. Therefore, model selection depends on the dominant requirement: maximum performance on clean data, operational efficiency, or tolerance to the controlled feature degradation considered

TABLE V

ROBUSTNESS TO CONTROLLED FEATURE-SPACE PERTURBATIONS, MEASURED BY MACRO-F1.

Model	Noise 0	Noise 0.01	Noise 0.03	Noise 0.05	Noise 0.10	Noise 0.20	Drop
HistGradientBoosting	**0.9976**	0.8260	0.7033	0.6575	0.6012	0.4849	0.5127
Robust Hybrid Stack	0.9965	0.9894	**0.9894**	**0.9894**	**0.9637**	**0.9525**	0.0439
Random Forest	0.9965	0.9810	0.9545	0.9226	0.8685	0.7692	0.2272
Extra Trees	0.9929	**0.9906**	0.9671	0.9485	0.9133	0.8258	0.1671
Logistic Regression	0.9753	0.9730	0.9729	0.9718	0.9453	0.9218	0.0535
Linear SVM	0.9741	0.9752	0.9694	0.9681	0.9304	0.8752	0.0989
KNN	0.9585	0.9562	0.9585	0.9608	0.9549	0.9422	**0.0163**

TABLE VI

PERFORMANCE BY FEATURE GROUP USED IN ISOLATION.

Group	Features	Acc.	Macro-F1
x	22	0.9894	0.9894
Bandpower	11	0.9812	0.9812
Spectral	11	0.9459	0.9457
Bandratio	11	0.9424	0.9424
Feature	2	0.6788	0.6655
Transient	1	0.6800	0.6630
PSD	1	0.5835	0.5786
Size	1	0.4929	0.4593

TABLE VII

TOP FEATURES ACCORDING TO PERMUTATION IMPORTANCE.

Rank	Feature	Mean	Std.
1	`bandpower_1_1p5ghz`	0.0926	0.0049
2	`transient_start_idx`	0.0828	0.0032
3	`bandpower_4_5ghz`	0.0734	0.0030
4	`x_kurtosis`	0.0639	0.0032
5	`bandpower_0_0p5ghz`	0.0631	0.0063
6	`spectral_entropy`	0.0469	0.0041
7	`x_crest_factor`	0.0300	0.0032
8	`x_mean`	0.0255	0.0031
9	`bandratio_2p5_3ghz`	0.0161	0.0016
10	`x_zero_crossing_rate`	0.0081	0.0014

here. The feature-importance analysis confirms that the x, *bandpower*, and *spectral* groups concentrate most of the discriminative information.

V. CONCLUSION

This work compared classical models, tabular neural networks, a tabular Transformer, and the proposed *Robust Hybrid Stack* for RF-based drone controller classification using the *MPACT DroneRC RF Dataset*.

HistGradientBoosting achieved the best clean-data performance and calibration, with accuracy and Macro-F1 of 0.9976. Tabular ResNet offered the best operational trade-off, while the *Robust Hybrid Stack* achieved the highest robustness to controlled feature-space perturbations, maintaining a Macro-F1 of 0.9525 at the strongest evaluated level.

These results indicate that no single model dominates all criteria: tree-based models favor clean-data accuracy, compact models favor efficient deployment, and the proposed hybrid model favors tolerance to feature degradation. However, this robustness does not imply resilience to noise or interference in the original RF signal.

The main limitations are the absence of acquisition- or session-level separation and of a conventional stacking baseline without perturbation-aware meta-training. Future work will therefore investigate group-aware validation, stacking ablations, signal-level degradation, cross-domain evaluation, additional RF datasets, and computational optimization of the *Robust Hybrid Stack*.

REFERENCES

[1] "A comprehensive survey on radio frequency (rf) fingerprinting: Traditional approaches, deep learning, and open challenges," *Computer Networks*, vol. 219, p. 109455, 2022.

[2] M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, A. Erbad, "Rf-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database," *Future Generation Computer Systems*, vol. 100, pp. 86–97, 2019.

[3] M. S. Allahham, M. F. Al-Sa'd, A. Al-Ali, A. Mohamed, T. Khattab, A. Erbad, "Dronerf dataset: A dataset of drones for rf-based detection, classification and identification," *Data in Brief*, vol. 26, p. 104313, 2019.

[4] M. Ezuma, F. Erden, C. K. Anjinappa, O. Ozdemir, I. Guvenc, "Micro-uav detection and classification from rf fingerprints using machine learning techniques," in *2019 IEEE Aerospace Conference*, 2019, pp. 1–13.

[5] ——, "Detection and classification of uavs using rf fingerprints in the presence of wi-fi and bluetooth interference," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 60–76, 2020.

[6] Y. Fu Z. He, "Radio frequency signal-based drone classification with frequency domain gramian angular field and convolutional neural network," *Drones*, vol. 8, no. 9, p. 511, 2024.

[7] "Application of cnn deep learning model and cfar filtering technique for rf-based drone signal classification in noisy conditions," *Journal of Military Science and Technology*, vol. 96, no. 96, pp. 30–40, 2024.

[8] L. H. Kirakosyan, M. V. Navoyan, V. G. Melkonyan, A. S. Sardaryan, S. S. Sargsyan, "Real-time radio signal classification based on spectrograms," *Programming and Computer Software*, vol. 51, no. 7, pp. 463–471, 2025.

[9] L. P. A. Guedes, P. H. M. Guedes, R. Caminha, "Drone classification from rf signals: A comparative study of convolutional networks and attention mechanisms," in *Proceedings of SIGE*, ITA, Brazil, 2025, pp. 022–027, iSSN: 1983-7402.

[10] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[11] P. Geurts, D. Ernst, L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[12] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[13] Y. Gorishniy, I. Rubachev, V. Khrulkov, A. Babenko, "Revisiting deep learning models for tabular data," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 18 932–18 943.

[14] K. He, X. Zhang, S. Ren, J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] D. H. Wolpert, "Stacked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–259, 1992.

[17] T. M. Cover P. E. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[18] C. Guo, G. Pleiss, Y. Sun, K. Q. Weinberger, "On calibration of modern neural networks," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 70, 2017, pp. 1321–1330.

[19] B. Efron, "Bootstrap methods: Another look at the jackknife," *The Annals of Statistics*, vol. 7, no. 1, pp. 1–26, 1979.

[20] C. M. Bishop, "Training with noise is equivalent to tikhonov regularization," *Neural Computation*, vol. 7, no. 1, pp. 108–116, 1995.