

## Robustness of Bayesian-Optimized Random Forest Against Feature-Space Evasion Attacks for Spam Detection

Moh. Aminudin<sup>1</sup>, Riza Arifudin<sup>2\*</sup>

<sup>1,2</sup>Computer Science Department, Faculty of Mathematics and Natural Sciences,  
Universitas Negeri Semarang, Indonesia

**Abstract.** Email spam filters may perform well on unmodified messages yet remain vulnerable to deliberate feature manipulation. Although hyperparameter optimization can improve clean-data performance, such gains do not establish robustness under test-time perturbations. A paired, leakage-controlled evaluation is therefore needed to assess both properties under identical conditions.

**Purpose:** This study compared the clean classification performance and feature-space robustness of a library-default Random Forest (RF-default) and a Bayesian-optimized Random Forest (BO-RF) on Spambase.

**Methods/Study design/approach:** Exact duplicates were removed, and predictor-identical records were grouped to prevent cross-fold leakage. Evaluation used two repetitions of five-fold nested stratified group-aware cross-validation with shared outer folds. Bayesian Optimization maximized inner-validation spam-class F1. Frozen outer-test folds were assessed on clean inputs, class-center-directed perturbations at  $\alpha = 0.05, 0.10, 0.20$ , and  $0.40$ , and 30 magnitude-matched random directions. Retained inference assessed within-model clean-versus-directed and directed-versus-random contrasts in F1, Recall, and model-specific attack success rate (ASR).

**Result/Findings:** BO-RF yielded slightly higher means across all four clean metrics; clean F1 was  $93.65 \pm 0.88\%$  versus  $93.53 \pm 0.98\%$  for RF-default. Bayesian Optimization selected the effective default configuration in nine of ten outer folds. At directed  $\alpha = 0.40$ , F1 fell to  $85.20\%$  for RF-default and  $85.31\%$  for BO-RF, while Recall fell to  $77.94\%$  and  $78.03\%$ , respectively; directed ASR was approximately  $15.5\%$  for both models. All 48 retained within-model contrasts met Holm-adjusted thresholds, indicating degradation relative to clean inputs and a stronger effect of directed than magnitude-matched random perturbations. No direct between-model inferential family was retained, so robustness comparisons remain descriptive.

**Novelty/Originality/Value:** Under the evaluated Spambase feature-space setting, clean-F1 Bayesian Optimization yielded only marginal descriptive gains and did not demonstrate a clear robustness advantage over the effective RF-default configuration. The findings show that clean-metric optimization should not be treated as a proxy for evasion robustness.

**Keywords:** Email spam detection, Random Forest, Bayesian Optimization, adversarial robustness, feature-space evasion attacks

**Received** August 2026 / **Revised** September 2026 / **Accepted** September 2026

This work is licensed under a [Creative Commons Attribution 4.0 International License](https://creativecommons.org/licenses/by/4.0/).



### INTRODUCTION

Machine-learning spam filters operate in a setting where message distributions and sender strategies can change [1]. Jáñez-Martino et al. [1] identify dataset shift and the presence of an active adversarial spammer as two central challenges in spam-email detection and review strategies used to contaminate or evade filters. In adversarial classification, an evasion attack modifies samples at test time to induce classification errors without changing the trained classifier [2], [3]. Clean predictive performance and performance under deliberate test-time manipulation therefore address different evaluation conditions; security-evaluation frameworks assess degradation under an explicitly stated attacker goal, knowledge, and capability [4].

Random Forest (RF) aggregates randomized tree predictors to produce an ensemble prediction [5]. Recent spam-detection studies demonstrate the predictive utility of RF-based pipelines but also reveal a shared evaluation limitation. Bouke et al. [6] emphasized class balancing, Gini-based feature selection, and post-hoc interpretability, whereas Omotehinwa and Oyewola [7] examined baseline and grid-searched ensemble models on the Enron1 corpus. Despite their different methodological emphases, both studies evaluated performance primarily through conventional clean-data classification metrics; neither isolated the effect of

---

\*Corresponding author.

Email addresses: [rizaarifudin@mail.unnes.ac.id](mailto:rizaarifudin@mail.unnes.ac.id) (Arifudin)

DOI: 10.15294/rji.v4i2.62732

RF hyperparameter optimization on robustness under controlled test-time perturbations. Consequently, their results do not establish whether gains in clean predictive performance translate into greater resistance to evasion or whether an optimized RF differs meaningfully from its effective default configuration under identical attack conditions. This shared limitation motivates a paired evaluation in which data partitions, perturbations, and eligible attack sets are explicitly controlled.

RF behavior can vary with hyperparameters such as the number of trees, the number of candidate features at a split, and node-size controls [8]. Bayesian Optimization provides a sample-efficient framework for expensive black-box search by using a probabilistic surrogate and an acquisition function to select subsequent configurations [9], [10]. However, the properties obtained through optimization are tied to the objective and search space supplied to the optimizer. In the present protocol, Bayesian Optimization maximized inner-validation spam-class F1 on clean data and contained no robustness term; consequently, any robustness advantage must be demonstrated empirically rather than inferred from the tuning procedure.

Adversarial robustness research in natural language processing distinguishes attacks and defenses across character, word, sentence, and representation levels and emphasizes that robustness conclusions depend on the perturbation and evaluation setting [11]. In spam detection, Cheng et al. [12] mapped feature perturbations to modified email content, Hotoğlu et al. [13] evaluated deep-learning filters under attacks at several textual levels, and Yang et al. [14] evaluated Random Forest on Spambase using robustness-oriented weighted training and inference-time randomization. Robustness evaluation must also distinguish feature-space manipulations from feasible problem-space attacks [15]. In a complementary defense-oriented direction, Taha [16] incorporated adversarial and multi-objective training into an email-spam framework. Collectively, these studies indicate that robustness gains are generally pursued through attack-aware objectives, altered training, randomized inference, or representation-level interventions, rather than through hyperparameters selected solely for clean predictive performance.

Taken together, the closest clean-tuning and robustness studies therefore answer different questions. Omotehinwa and Oyewola [7] reported a tuned RF F1 score of 97.85% on Enron1 dataset after grid-search optimization but did not evaluate test-time evasion; Yang et al. [14] evaluated RF evasion on Spambase but did not compare a version-pinned RF-default with BO-RF selected using clean spam-class F1; and Taha [16] incorporated adversarial objectives into the training framework. To the best of our knowledge, within the scope of the literature reviewed, prior work has not sufficiently examined a version-pinned effective RF-default configuration and the same RF classifier selected solely through clean-data Bayesian Optimization within a single duplicate-aware, leakage-controlled protocol using shared outer folds, identical directed feature-space perturbations, magnitude-matched random controls, and model-specific eligible sets fixed from clean predictions. It therefore remains unclear whether clean-metric Bayesian Optimization improves, degrades, or leaves unchanged feature-space robustness under this controlled evaluation. This unresolved relationship is not merely the combination of Random Forest, Bayesian Optimization, and feature-space perturbation testing defines the empirical gap addressed in this study.

Accordingly, this study compares the Random Forest configuration selected through Bayesian Optimization (BO-RF) with the effective library-default configuration (RF-default) in clean classification and feature-space robustness on Spambase. The robustness analysis examines within-model changes in spam F1, Recall, and model-specific ASR under class-center-directed perturbations and contrasts the directed perturbations with magnitude-matched random-direction controls. Both models receive identical evaluation folds and perturbation arrays.

This study makes three contributions. First, it provides empirical evidence that clean-F1 Bayesian Optimization produced only marginal descriptive gains and did not demonstrate a clear feature-space robustness advantage over the effective RF-default configuration. Second, it isolates this relationship through duplicate-aware nested evaluation with group-constrained folds, shared outer folds, identical perturbation arrays, magnitude-matched random controls, and model-specific attack success rates. Third, it distinguishes within-model attack effects from descriptive between-model differences, so the evidence is not overstated as a formal BO-RF-versus-RF-default equivalence test. Because Spambase contains aggregate numerical attributes rather than raw messages [17], the contributions are restricted to the stated dataset, Random Forest implementation, optimization budget, and feature-space threat model.

## METHODS

This study used a paired quantitative experiment to compare an untuned Random Forest (RF-default) with a Bayesian-optimized Random Forest (BO-RF) on clean inputs and under controlled feature-space perturbations. The protocol covered Spambase, a shared class-center-directed perturbation, and a magnitude-matched random control. The following subsections describe data preparation, model construction, optimization, attack design, evaluation, and the nested experimental workflow.

### Data Collection

The experiments used Spambase from the UCI Machine Learning Repository [17]. The repository documents 4,601 email records, 57 predictors, and one binary outcome [17]. Its schema divides the predictors into 48 word-frequency attributes, six character-frequency attributes, and three capitalization-run attributes, and codes spam as 1 and non-spam as 0 [17]. The comma-delimited data file and repository schema were used without external records or engineered predictors.

### Data Preprocessing

Before fold construction, the raw file was checked against the repository schema for row width, numeric predictors, binary labels, missing and non-finite values, non-negative feature domains, and duplicate records. Exact duplicate rows, including the label, were removed using a deterministic keep-first rule. Records with identical 57-predictor vectors were assigned to the same group and kept within one partition during every group-aware split, including pairs with conflicting labels. This prevented predictor-identical observations from appearing in both training and evaluation partitions.

No imputation, resampling, synthetic oversampling, or class reweighting was applied. Predictors were scaled to  $[0, 1]$  using `MinMaxScaler` with `clip = True` [18]. The scaler was fitted only on the corresponding training partition: separately within each inner-training fold during optimization and then on the complete outer-training partition for final evaluation. Both models and all clean and perturbed conditions within an outer fold used the same fitted scaler.

### Random Forest Baseline

Random Forest followed Breiman [5] and was implemented using `RandomForestClassifier` from `scikit-learn` 1.8.0 [18]. RF-default used the effective defaults of the pinned library version, with `random_state = 2026080721` and `n_jobs = 1` set explicitly for reproducibility. Thus, default denotes the configuration of the pinned library version. No resampling or class weighting was applied, and spam was treated as the positive class.

Within each outer fold, RF-default was fitted on the preprocessed outer-training partition. After hyperparameter selection, BO-RF was refitted on the same observations using its selected configuration and the same RF seed. Both models were evaluated on identical clean and perturbed outer-test samples. A sample was classified as spam ( $\hat{y} = 1$ ) when  $p(y = 1) \geq 0.5$  and as non-spam otherwise. The paired contrast therefore held the partitions, preprocessing, RF seed, classification threshold, and test arrays constant while changing the RF configuration. The complete effective RF-default configuration is presented in Table 1.

Table 1. Effective RF-default configuration

| Parameter         | Effective value |
|-------------------|-----------------|
| n_estimators      | 100             |
| Criterion         | gini            |
| max_depth         | None            |
| min_samples_split | 2               |
| min_samples_leaf  | 1               |
| max_features      | sqrt            |
| Bootstrap         | True            |
| oob_score         | False           |
| class_weight      | None            |
| max_samples       | None            |
| random_state      | 2026080721      |
| n_jobs            | 1               |

### Bayesian Optimization of Random Forest

BO-RF hyperparameters were selected independently within each outer-training partition using bayesian-optimization 3.3.0 with seed 2026080731 [19]. Bayesian Optimization used a Gaussian-process surrogate

and previous evaluations to select configurations for the inner-validation objective [9]. The upper-confidence-bound acquisition function used  $\kappa = 2.576$  [20]. The search space is summarized in Table 2.

Table 2. Bayesian Optimization search space

| Hyperparameter    | Type        | Search values                     | Fixed/default-relevant value |
|-------------------|-------------|-----------------------------------|------------------------------|
| n_estimators      | Integer     | 100–500                           | 100 included                 |
| max_depth         | Categorical | None, 5, 10, 15, 20, 30, 40       | None included                |
| min_samples_split | Integer     | 2–20                              | 2 included                   |
| min_samples_leaf  | Integer     | 1–10                              | 1 included                   |
| max_features      | Categorical | sqrt, log2, 0.25, 0.50, 0.75, 1.0 | sqrt included                |
| criterion         | Fixed       | Gini                              | gini                         |
| bootstrap         | Fixed       | True                              | True                         |
| class_weight      | Fixed       | None                              | None                         |

Candidate configurations were assessed on fixed five-fold stratified group-aware inner splits within each outer-training partition [18]. Feature-vector groups were supplied to every split, and the same inner folds were reused for all candidates in an outer fold. The objective was mean spam-class F1 across the five validation folds, with scaling fitted separately on each inner-training split. Restricting selection to outer-training data avoids selection-induced optimism in outer evaluation [21]. The budget comprised 55 unique effective configurations: the RF-default point, four randomly initialized points, and 50 UCB-guided points. Repeated effective configurations did not consume the unique budget. The configuration with the highest mean inner-fold spam F1 was selected; ties were resolved by lower inner-fold standard deviation and then earlier evaluation order. The selected BO-RF was refitted on the complete outer-training partition.

### Adversarial Evasion Attack Design

The robustness assessment was a feature-space evasion stress test and did not claim that an actionable raw email had been generated. Attacker goal, knowledge, and capability were specified following adversarial-classification evaluation frameworks [2], [3], [4] and are summarized in Table 3.

Table 3. Feature-space threat model

| Component           | Specification                                                                                                                                                                                                                                                                                |
|---------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Attack timing       | Test-time evasion after both models and preprocessing are fitted                                                                                                                                                                                                                             |
| Attacker goal       | Cause ground-truth spam (label 1) to be predicted as non-spam (label 0)                                                                                                                                                                                                                      |
| Knowledge           | Model-agnostic, data-informed: scaled outer-training class centers and feature domain; no model parameters, gradients, or model feature importance. Outer-test labels identify ground-truth spam perturbation targets only; they are not used to select features or perturbation directions. |
| Perturbation target | Ground-truth spam in the outer-test fold; non-spam samples and all labels remain unchanged                                                                                                                                                                                                   |
| Feature capability  | Ten shared predictors selected from outer-training class-center separation                                                                                                                                                                                                                   |
| Attack set          | Identical directed and random-control arrays supplied to RF-default and BO-RF                                                                                                                                                                                                                |
| Domain constraints  | Projection to [0, 1], plus inverse-scale/round/rescale handling for integer-derived count features                                                                                                                                                                                           |
| Interpretive scope  | Feature-space robustness stress test only; no guarantee of semantic, syntactic, or delivery feasibility in a raw email                                                                                                                                                                       |

For each outer fold, class centers were computed from the scaled outer-training data only. Let  $\mu_{1j}$  and  $\mu_{0j}$  denote the mean of feature  $j$  for spam and non-spam, respectively. Features were ranked by absolute class-center separation, with ties resolved by ascending zero-based feature index. This ranking was a prespecified, study-specific construction rather than an algorithm attributed to an external source. The separation score and selection of the ten highest-ranked predictors are formalized in Equation (1).

$$q_j = |\mu_{1j} - \mu_{0j}|, \quad S = \text{TopK}_j(q_j), \quad k = 10 \quad (1)$$

For each ground-truth spam sample  $i$  and selected feature  $j$ , the directed perturbation moved the scaled value toward the non-spam class center; unselected features, non-spam samples, and labels were unchanged. The corresponding perturbation rule is formalized in Equation (2).

$$\delta_{\text{dir},ij} = \alpha(\mu_{0j} - x_{ij}), \quad x'_{\text{dir},ij} = \Pi_{[0,1]}(x_{ij} + \delta_{\text{dir},ij}), \quad j \in S \quad (2)$$

Here,  $\Pi_{[0,1]}$  denotes projection onto the scaled domain. The four nonzero perturbation levels were  $\alpha \in \{0.05, 0.10, 0.20, 0.40\}$ ;  $\alpha = 0$  represented clean evaluation. Ranking, class centers, and perturbation arrays were

shared by both models, so a model-specific feature-importance advantage could not enter attack construction.

A magnitude-matched random control perturbed the same ground-truth spam targets, selected features, and coordinate-wise nominal magnitudes. The random-control perturbation and its equiprobable sign assignment for random direction  $r$  are formalized in Equation (3).

$$\begin{aligned} \delta_{\text{rand},ij}^{(r)} &= s_{ij}^{(r)} |\delta_{\text{dir},ij}|, & x'_{\text{rand},ij} &= \Pi_{[0,1]}(x_{ij} + \delta_{\text{rand},ij}^{(r)}) \\ s_{ij}^{(r)} &\in \{-1, +1\}, & P(s_{ij}^{(r)} = -1) &= P(s_{ij}^{(r)} = +1) = \frac{1}{2} \end{aligned} \quad (3)$$

Thirty random sign matrices were generated per  $\alpha$  and outer fold with seed 2026080741, and identical perturbation arrays were supplied to RF-default and BO-RF. Accuracy, spam-class Precision, Recall, F1, and the confusion matrix were evaluated on the complete outer-test fold, whereas ASR used the eligible set defined below. Random-control metrics were first averaged across the 30 directions within each outer fold. The resulting fold-level F1, Recall, and ASR values were used for the retained paired analysis, preserving the outer fold as the inferential unit. If capitalization-run count features were selected, perturbed values were inverse-transformed, rounded to the nearest non-negative integer, transformed again with the outer-training scaler, and clipped to the scaled domain [17]. This numerical handling does not establish plausible raw email because feature-space modifications require additional semantic and problem-space feasibility constraints [12], [15].

#### Evaluation Metrics and Statistical Analysis

Accuracy, spam-class Precision, Recall, F1, and the confusion matrix were evaluated on every outer-test fold under clean and perturbed conditions using standard classification definitions [22]. Attack success rate and relative F1 degradation were additionally used to characterize perturbation effects. Predicted labels used  $p(y = 1) \geq 0.5$ , with spam as the positive class. Clean Accuracy, Precision, Recall, and F1 were summarized descriptively. The retained inferential endpoints were F1, Recall, and model-specific ASR under directed perturbations and their magnitude-matched random controls; Accuracy, Precision, the confusion matrix, and relative F1 degradation remained descriptive outcomes.

For model  $m$  in outer fold  $f$ , ASR was defined on  $C_{m,f}$ , the ground-truth spam correctly classified under clean evaluation. This set was fixed from clean predictions and reused across all directed conditions and 30 random-control directions at each  $\alpha$ . Model-specific ASR for attack type  $a$  and perturbation level  $\alpha$  is defined in Equation (4).

$$\text{ASR}_{m,f}^{(a,\alpha)}(C) = \frac{1}{|C|} \sum_{i \in C} \mathbf{1}[\hat{y}_{m,i,f}^{(a,\alpha)} = 0] \times 100\% \quad (4)$$

Perturbed predictions did not change eligible-set membership. Model-specific ASR therefore measured the proportion of initially correct spam predictions that changed to non-spam under the evaluated perturbation. Absolute and relative F1 degradation from clean evaluation were calculated as shown in Equation (5).

$$D_{\text{abs}} = M_{\text{clean}} - M_{\text{attack}}, \quad D_{\text{rel}} = \frac{D_{\text{abs}}}{M_{\text{clean}}} \times 100\% \quad (5)$$

Absolute change is reported in percentage points, whereas relative F1 degradation is reported as a percentage of the clean F1 value. For directed-versus-random inference, each fold's directed F1, Recall, and ASR were paired with the corresponding within-fold mean across 30 random directions.

The retained hypotheses used  $J = 10$  paired outer-fold evaluations and two-sided corrected repeated-fold paired t-tests. Let  $\bar{d}$  denote the mean paired difference,  $sd$  its sample standard deviation, and  $\bar{r}$  the observed mean outer-test-to-outer-training size ratio. Following the corrected repeated-fold procedure [23], [24], the corrected standard error and 95% confidence interval were calculated using Equation (6).

$$SE_{\text{corr}} = \sqrt{\left(\frac{1}{J} + \bar{r}\right) s_d^2}, \quad CI_{95\%} = \bar{d} \pm t_{J-1, 0.975} SE_{\text{corr}} \quad (6)$$

The corrected test statistic was  $\bar{d}$  divided by the corrected standard error. The significance level was 0.05. Holm's sequential adjustment [25] was applied separately within each retained comparison family. The first retained family compared clean and direct conditions separately within each model and  $\alpha$  for F1, Recall, and model-specific ASR (2 models  $\times$  4  $\alpha$  levels  $\times$  3 endpoints = 24 hypotheses). The second compared directed conditions with within-fold random means for the same models,  $\alpha$  levels, and endpoints (24 hypotheses). No other inferential comparisons were included in the article.

### Experimental Design and Workflow

Outer evaluation used two independently seeded five-fold StratifiedGroupKFold procedures from scikit-learn 1.8.0 [18] (shuffle=True, seeds 2026080701 and 2026080702), yielding ten paired evaluations. Each observation appeared once in an outer-test fold, with identical feature-vector groups kept together. The outer-test fold was sequestered during inner group-aware validation for BO-RF selection. The complete outer-training partition determined the scaler, fitted models, class centers, and top-ten attack features, which were frozen before outer-test evaluation. Clean, class-center-directed, and magnitude-matched random test arrays were prepared once per fold and applied identically to RF-default and BO-RF.

No outer-test information influenced inner-fold scaling, hyperparameter selection, class-center estimation, or attack-feature ranking. Within each outer fold, the scaler and both Random Forest models were fitted on the complete outer-training partition before outer-test prediction. Shared folds and identical evaluation arrays ensured paired RF-default and BO-RF observations. Between-model differences were interpreted descriptively, while inference was restricted to within-model clean-versus-directed and directed-versus-random contrasts. Fixed seeds, indices, configurations, perturbation settings, predictions, and fold-level metrics were retained for reproducibility. The nested group-aware workflow is summarized in Figure 1.

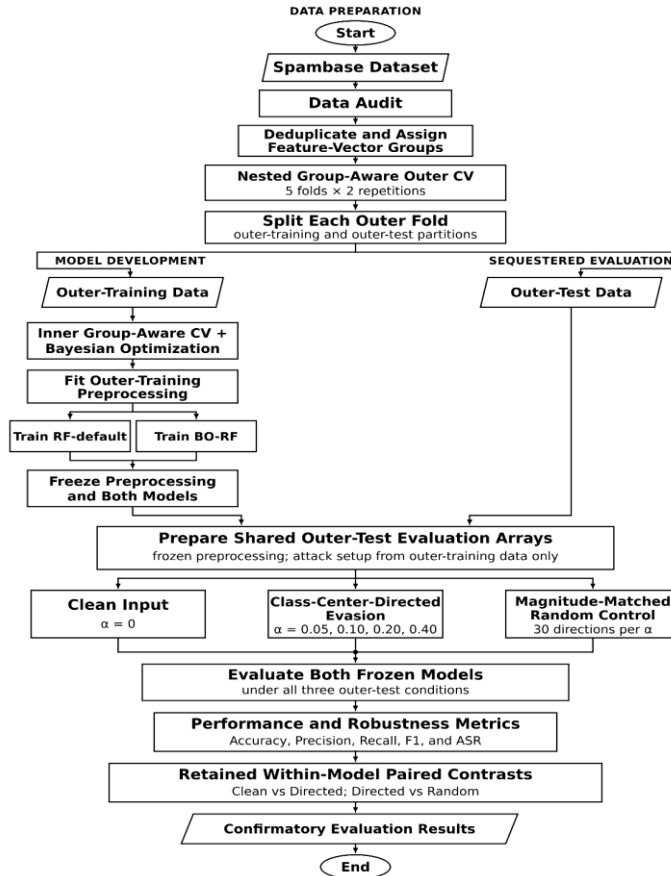


Figure 1. Nested group-aware experimental workflow.

## RESULT AND DISCUSSION

### Dataset Audit and Analytic Sample

The raw file matched the documented 57-predictor schema and contained no missing, non-finite, or out-of-domain values [17]. Deterministic keep-first removal eliminated 391 exact duplicate rows, leaving 4,210 analytic observations: 2,531 non-spam and 1,679 spam. Three repeated predictor-vector groups (six rows) remained, and all three contained conflicting labels. These rows were retained because their labels could not be adjudicated from the repository; assigning a shared group identifier kept each conflicting pair within one-fold partition. Table 4 reports the complete audit outcome.

Table 4. Dataset audit and analytic sample

| Audit item                       | Outcome           |
|----------------------------------|-------------------|
| Raw records                      | 4,601             |
| Predictors / outcome             | 57 / 1            |
| Raw non-spam / spam              | 2,788 / 1,813     |
| Missing / non-finite values      | 0 / 0             |
| Exact duplicate rows removed     | 391               |
| Analytic records                 | 4,210             |
| Analytic non-spam / spam         | 2,531 / 1,679     |
| Repeated predictor-vector groups | 3 groups / 6 rows |
| Conflicting-label groups         | 3 groups / 6 rows |
| Feature-domain violations        | 0                 |

All ten precomputed outer folds contained both classes, had no sample or group overlap, and assigned every observation to outer-test exactly once per repeat. Absolute fold-level deviation from the overall spam proportion ranged from 0.000236 to 0.000950. The observed mean outer-test-to-outer-training ratio was 0.2500001 and was retained as empirical  $\bar{r}$  in the corrected inference.

### Confirmatory Run and Optimization Outcome

The confirmatory run completed all 10 outer evaluations in 01:53:30. Outer-training partitions contained 3,367-3,369 observations and outer-test partitions contained 841-843. The stored fold summaries recorded no fit warnings or failed inner fits. Selected inner-validation mean F1 ranged from 92.774% to 93.483%, with fold-level standard deviations of 0.573-1.681 percentage points.

Bayesian Optimization selected the effective RF-default configuration in 9 of 10 outer folds. Repeat 2, fold 3 selected a distinct configuration with 427 trees, maximum depth 30, minimum split size 3, minimum leaf size 1, and log2 candidate features; the fixed settings remained Gini splitting, bootstrapping, no class weighting, `n_jobs` = 1, and `random_state` = 2026080721. The final package records total run time but does not provide per-fold wall-clock times or duplicate-proposal counts.

### Clean Classification Performance

Accuracy reports overall classification correctness; spam-class Precision reports how often predictions labelled as spam are correct; spam-class Recall reports how much spam remains detected; and F1 balances Precision and Recall. BO-RF achieved slightly higher means than RF-default for all four clean metrics: Accuracy was  $95.01 \pm 0.66\%$  versus  $94.92 \pm 0.74\%$ , Precision was  $95.07 \pm 0.85\%$  versus  $94.95 \pm 0.94\%$ , Recall was  $92.29 \pm 1.79\%$  versus  $92.17 \pm 1.85\%$ , and F1 was  $93.65 \pm 0.88\%$  versus  $93.53 \pm 0.98\%$ . Table 5 summarizes these results. Values are mean  $\pm$  sample standard deviation across the 10 outer folds.

Table 5. Clean classification performance

| Model      | Accuracy (%)     | Precision (%)    | Recall (%)       | F1-score (%)     |
|------------|------------------|------------------|------------------|------------------|
| RF-default | $94.92 \pm 0.74$ | $94.95 \pm 0.94$ | $92.17 \pm 1.85$ | $93.53 \pm 0.98$ |
| BO-RF      | $95.01 \pm 0.66$ | $95.07 \pm 0.85$ | $92.29 \pm 1.79$ | $93.65 \pm 0.88$ |

Figure 2 presents the pooled out-of-fold clean confusion matrices for both models.

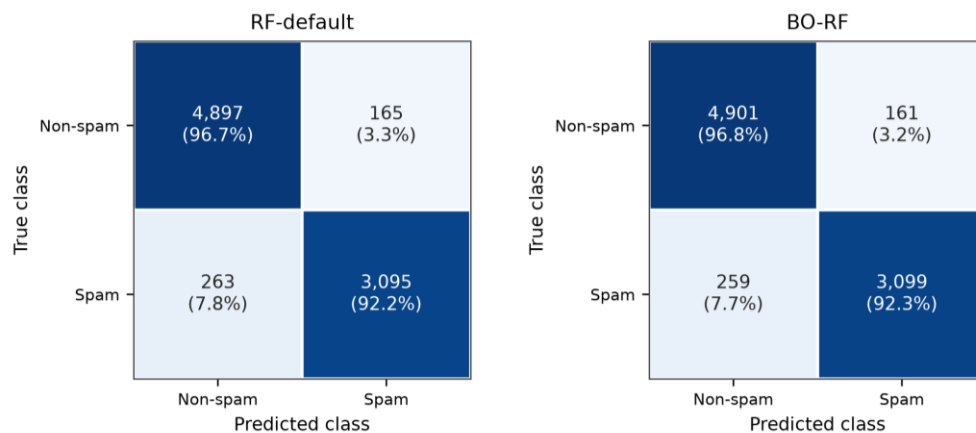


Figure 2. Comparison of clean confusion matrix results

RF-default produced 4,897 true negatives, 165 false positives, 263 false negatives, and 3,095 true positives. BO-RF produced 4,901 true negatives, 161 false positives, 259 false negatives, and 3,099 true positives. Counts were pooled from stored out-of-fold predictions across two cross-validation repetitions and normalized within the true class for display.

Figure 3 shows the paired clean F1 distributions across the 10 outer evaluations.

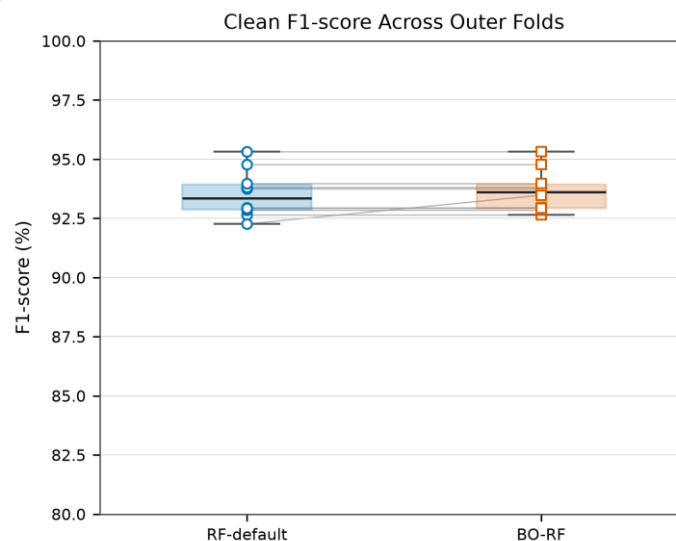


Figure 3. Clean F1-score across outer folds

Gray lines connect paired outer-fold results. The vertical axis is fixed at 80-100% so the closely aligned distributions are shown without an exaggerated scale.

### Robustness to Directed Evasion Attacks and Magnitude-Matched Random Controls

The robustness evaluation first considered actual post-perturbation classification performance. Clean rows provide the reference for each model, while random-control values were averaged across 30 directions within each outer fold before the 10 fold means were summarized. Table 6 reports Accuracy, Precision, Recall, and F1 for every evaluated condition.

Table 6. Classification performance under perturbation

| Model      | Perturbation | $\alpha$ | Accuracy (%) | Precision (%) | Recall (%) | F1-score (%) |
|------------|--------------|----------|--------------|---------------|------------|--------------|
| RF-default | Clean        | 0.00     | 94.92        | 94.95         | 92.17      | 93.53        |
| RF-default | Directed     | 0.05     | 94.45        | 94.89         | 91.01      | 92.90        |
| RF-default | Directed     | 0.10     | 93.80        | 94.80         | 89.37      | 91.99        |
| RF-default | Directed     | 0.20     | 91.45        | 94.44         | 83.47      | 88.61        |
| RF-default | Directed     | 0.40     | 89.24        | 94.04         | 77.94      | 85.20        |
| RF-default | Random       | 0.05     | 94.70        | 94.92         | 91.63      | 93.24        |
| RF-default | Random       | 0.10     | 94.40        | 94.88         | 90.88      | 92.82        |
| RF-default | Random       | 0.20     | 93.39        | 94.74         | 88.33      | 91.41        |
| RF-default | Random       | 0.40     | 92.51        | 94.61         | 86.13      | 90.16        |
| BO-RF      | Clean        | 0.00     | 95.01        | 95.07         | 92.29      | 93.65        |
| BO-RF      | Directed     | 0.05     | 94.49        | 95.00         | 90.98      | 92.94        |
| BO-RF      | Directed     | 0.10     | 93.87        | 94.92         | 89.43      | 92.08        |
| BO-RF      | Directed     | 0.20     | 91.56        | 94.58         | 83.62      | 88.76        |
| BO-RF      | Directed     | 0.40     | 89.32        | 94.18         | 78.03      | 85.31        |
| BO-RF      | Random       | 0.05     | 94.77        | 95.04         | 91.69      | 93.32        |
| BO-RF      | Random       | 0.10     | 94.49        | 95.01         | 90.99      | 92.94        |
| BO-RF      | Random       | 0.20     | 93.47        | 94.87         | 88.42      | 91.52        |
| BO-RF      | Random       | 0.40     | 92.57        | 94.73         | 86.16      | 90.23        |

Values are means across 10 outer folds. Random-control metrics were first averaged across 30 directions within each outer fold. All values are percentages except  $\alpha$ , which is unitless. The results demonstrate that the directed perturbations successfully degraded classification performance within the evaluated feature-space setting. At  $\alpha = 0.40$ , RF-default Accuracy decreased from 94.92% on clean inputs to 89.24%, while BO-RF Accuracy decreased from 95.01% to 89.32%. Recall showed a greater reduction, decreasing from 92.17% to 77.94% for RF-default and from 92.29% to 78.03% for BO-RF. Consequently, F1 decreased from 93.53% to 85.20% for RF-default and from 93.65% to 85.31% for BO-RF. Precision changed more moderately, decreasing from 94.95% to 94.04% for RF-default and from 95.07% to 94.18% for BO-RF. The greater reduction in Recall indicates that the directed perturbations primarily increased false negatives, allowing more spam messages to be classified as non-spam.

At directed  $\alpha = 0.40$ , the Accuracy reductions were 5.68 and 5.69 percentage points for RF-default and BO-RF, respectively; the corresponding Recall reductions were 14.23 and 14.26 percentage points, and the F1 reductions were 8.33 and 8.34 percentage points. Relative F1 degradation reached 8.915% for both models, while directed ASR reached 15.470% for RF-default and 15.484% for BO-RF. Percentage-point changes and relative degradation are reported separately because they use different denominators. BO-RF consistently achieved slightly higher post-perturbation Accuracy than RF-default across all four directed and all four random-control conditions, while both models exhibited the same general degradation pattern as  $\alpha$  increased. At directed  $\alpha = 0.40$ , BO-RF retained 89.32% Accuracy compared with 89.24% for RF-default. These between-model values are descriptive. Figure 4 shows model-specific ASR as perturbation magnitude increases.

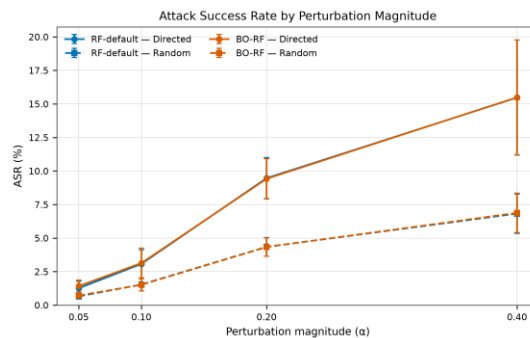


Figure 4. Attack success rate across perturbation levels

Directed ASR increased from 1.255% at  $\alpha = 0.05$  to 15.470% at  $\alpha = 0.40$  for RF-default and from 1.417% to 15.484% for BO-RF. The matched random-control values increased from 0.649% to 6.818% and from 0.712% to 6.876%, respectively. Directed ASR was higher than random ASR at every  $\alpha$ , and the difference widened as perturbation magnitude increased. Figure 5 presents relative F1 degradation under the same conditions.

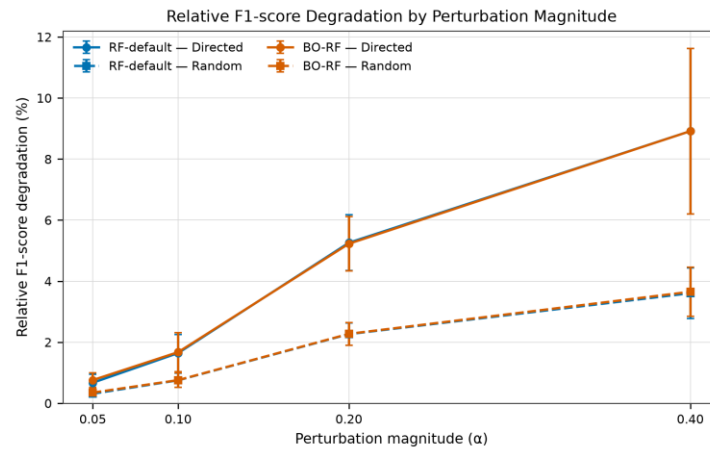


Figure 5. Relative F1-score degradation across perturbation levels

Directed relative F1 degradation increased from 0.672% to 8.915% for RF-default and from 0.759% to 8.915% for BO-RF between  $\alpha = 0.05$  and 0.40. The corresponding random-control degradation increased from 0.312% to 3.605% and from 0.349% to 3.654%. Thus, directed perturbations produced greater F1 degradation than the magnitude-matched controls at every evaluated level.

### Statistical Analysis

All 48 retained hypotheses were estimable. Table 7 summarizes the two separately adjusted 24-hypothesis comparison families and their observed directions.

Table 7. Summary of the retained confirmatory statistical analysis

| Comparison family  | Endpoints                          | Meeting threshold | Holm-adjusted p-value range | Observed direction                                                                                         |
|--------------------|------------------------------------|-------------------|-----------------------------|------------------------------------------------------------------------------------------------------------|
| Clean vs directed  | F1, Recall, and model-specific ASR | 24/24             | 0.0000526 to 0.0092483      | Directed perturbations reduced F1 and Recall and produced ASR above the clean reference condition.         |
| Directed vs random | F1, Recall, and model-specific ASR | 24/24             | 0.0001696 to 0.0426664      | Directed perturbations produced lower F1 and Recall and higher ASR than magnitude-matched random controls. |

Note: Tests used two-sided corrected repeated-fold paired t-tests with Holm adjustment applied separately within each 24-hypothesis family.

For the retained endpoints, all 24 clean-to-directed comparisons involving F1, Recall, and model-specific ASR met the Holm-adjusted threshold; adjusted p-values ranged from 0.0000526 to 0.0092483. All 24 directed-versus-random comparisons for the same endpoints also met the adjusted threshold, with adjusted p-values from 0.0001696 to 0.0426664. The retained confirmatory analysis supports the directed attack effect on F1, Recall, and ASR and shows that perturbation direction contributed beyond magnitude alone.

### Comparative Discussion

The clean-data results indicate only a marginal descriptive advantage for BO-RF: its mean Accuracy, Precision, Recall, and F1 were slightly higher than those of RF-default. This pattern should be interpreted considering the optimization outcome, because Bayesian Optimization selected the effective default configuration in nine of the ten outer folds and selected a distinct configuration only in the remaining fold. Under the stated search space and clean-F1 objective, optimization therefore produced limited descriptive separation from the effective RF-default configuration on clean data.

Under perturbation, the dominant observed error pattern was a shift of spam cases from true-positive to false-negative predictions: Recall declined more strongly than Precision as  $\alpha$  increased. This pattern is qualitatively consistent with the feature-space email evasion examined by Cheng et al. [12] and the Random Forest evasion setting studied by Yang et al. [14], in which successful perturbations correspond to spam being classified as non-spam. Direct numerical comparisons with those studies are not warranted because their feature representations, attack constructions, and evaluation protocols differ.

The directed-versus-random contrasts indicate that perturbation direction contributed to degradation beyond the effect associated with perturbation magnitude alone. Shared outer folds, shared preprocessing, and identical perturbation arrays reduced avoidable sources of variation, supporting a controlled descriptive comparison between RF-default and BO-RF. Taken together, the evidence supports evaluating clean-data optimization and feature-space robustness jointly while distinguishing descriptive between-model patterns from the retained within-model inferential contrasts.

### Limitations

The evidence is limited to one dataset, Spam base, its aggregate 57-feature numerical representation, the evaluated Random Forest implementation, and the stated Bayesian Optimization budget. The class-center heuristic perturbed ten selected features, and 30 random directions provided a prespecified magnitude-matched control rather than an exhaustive perturbation distribution. Most importantly, the attack operates in normalized feature space; projection and integer handling do not guarantee that a perturbed vector can be converted into a semantically coherent, syntactically valid, and deliverable email. Future work should evaluate additional spam datasets, realizable raw-email problem-space attacks with semantic and feasibility constraints, other model families, and robustness-aware optimization objectives.

### CONCLUSION

Under the evaluated Spam base feature-space setting, clean-F1 Bayesian Optimization yielded only marginal descriptive gains over the effective RF-default configuration and did not demonstrate a clear robustness advantage. The optimizer selected the effective default in nine of ten outer folds, while both models showed descriptively similar degradation under increasingly strong directed perturbations, mainly through lower Recall and more false negatives. Directed perturbations were associated with stronger degradation than magnitude-matched random controls, indicating that perturbation direction contributed beyond magnitude alone. Because the retained inferential families assessed within-model attack effects rather than a direct BO-RF-versus-RF-default hypothesis, this between-model robustness conclusion remains descriptive. Clean-data optimization should therefore be paired with attack-specific robustness evaluation rather than treated as evidence of improved evasion robustness. These findings are limited to the evaluated dataset, Random Forest implementation, optimization budget, and feature-space threat model; future work should examine realizable raw-email attacks, broader datasets and models, and robustness-aware optimization objectives.

### REFERENCES

- [1] F. Jáñez-Martino, R. Alaiz-Rodríguez, V. González-Castro, E. Fidalgo, and E. Alegre, "A review of spam email detection: analysis of spammer strategies and the dataset shift problem," *Artif. Intell. Rev.*, vol. 56, no. 2, pp. 1145–1173, Feb. 2023, doi: 10.1007/s10462-022-10195-4.
- [2] B. Biggio et al., "Evasion Attacks against Machine Learning at Test Time," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8190 LNAI, no. PART 3, Springer, Berlin, Heidelberg, 2013, pp. 387–402. doi: 10.1007/978-3-642-40994-3\_25.
- [3] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognit.*, vol. 84, pp. 317–331, Dec. 2018, doi: 10.1016/j.patcog.2018.07.023.
- [4] B. Biggio, G. Fumera, and F. Roli, "Security Evaluation of Pattern Classifiers under Attack," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 4, pp. 984–996, Apr. 2014, doi: 10.1109/TKDE.2013.57.
- [5] L. Breiman, "Random Forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.
- [6] M. A. Bouke, O. I. Alramli, and A. Abdullah, "XAIRF-WFP: a novel XAI-based random forest classifier for advanced email spam detection," *Int. J. Inf. Secur.*, vol. 24, no. 1, p. 5, Feb. 2025, doi: 10.1007/s10207-024-00920-1.
- [7] T. O. Omotehinwa and D. O. Oyewola, "Hyperparameter Optimization of Ensemble Models for Spam Email Detection," *Applied Sciences*, vol. 13, no. 3, p. 1971, Feb. 2023, doi: 10.3390/app13031971.
- [8] P. Probst, M. N. Wright, and A. Boulesteix, "Hyperparameters and tuning strategies for random forest," *WIREs Data Mining and Knowledge Discovery*, vol. 9, no. 3, p. e1301, May 2019, doi: 10.1002/widm.1301.
- [9] J. Snoek, H. Larochelle, and R. Adams, "Practical Bayesian Optimization of Machine Learning Algorithms," in *Advances in Neural Information Processing Systems*, F. Pereira, C. J. Burges, L. Bottou, and K. Weinberger, Eds., Curran Associates, Inc., 2012. [Online]. Available: [https://proceedings.neurips.cc/paper\\_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf](https://proceedings.neurips.cc/paper_files/paper/2012/file/05311655a15b75fab86956663e1819cd-Paper.pdf)

- [10] X. Wang, Y. Jin, S. Schmitt, and M. Olhofer, "Recent Advances in Bayesian Optimization," *ACM Comput. Surv.*, vol. 55, no. 13s, pp. 1–36, Dec. 2023, doi: 10.1145/3582078.
- [11] S. Goyal, S. Doddapaneni, M. M. Khapra, and B. Ravindran, "A Survey of Adversarial Defenses and Robustness in NLP," *ACM Comput. Surv.*, vol. 55, no. 14s, pp. 1–39, Dec. 2023, doi: 10.1145/3593042.
- [12] Q. Cheng, A. Xu, X. Li, and L. Ding, "Adversarial Email Generation against Spam Detection Models through Feature Perturbation," in *2022 IEEE International Conference on Assured Autonomy (ICAA)*, IEEE, Mar. 2022, pp. 83–92. doi: 10.1109/ICAA52185.2022.00019.
- [13] E. Hotoğlu, S. Sen, and B. Can, "A comprehensive analysis of adversarial attacks against spam filters," *Comput. Secur.*, in press, Art. no. 105066, available online Jul. 23, 2026, doi: 10.1016/j.cose.2026.105066.
- [14] F. Yang, Z. Chen, and A. Gangopadhyay, "Using Randomness to Improve Robustness of Tree-Based Models Against Evasion Attacks," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 2, pp. 969–982, Feb. 2022, doi: 10.1109/TKDE.2020.2987299.
- [15] F. Pierazzi, F. Pendlebury, J. Cortellazzi, and L. Cavallaro, "Intriguing Properties of Adversarial ML Attacks in the Problem Space," in *2020 IEEE Symposium on Security and Privacy (SP)*, IEEE, May 2020, pp. 1332–1349. doi: 10.1109/SP40000.2020.00073.
- [16] K. Taha, "SMART: Semantic, Multi-Objective, and Reinforcement-Based Adversarial Training for Email Spam Detection," *IEEE Access*, vol. 13, pp. 112749–112764, 2025, doi: 10.1109/ACCESS.2025.3581131.
- [17] M. Hopkins, E. Reeber, G. Forman, and J. Suermondt, "Spambase," UCI Machine Learning Repository. Accessed: May 29, 2026. [Online]. Available: <https://doi.org/10.24432/C53G6X>
- [18] F. Pedregosa *et al.*, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, no. 85, pp. 2825–2830, 2011, [Online]. Available: <http://jmlr.org/papers/v12/pedregosa11a.html>
- [19] F. Nogueira, "Bayesian Optimization: Open source constrained global optimization tool for Python," 2014, Accessed: Jun. 19, 2026. [Online]. Available: <https://github.com/bayesian-optimization/BayesianOptimization>
- [20] N. Srinivas, A. Krause, S. M. Kakade, and M. W. Seeger, "Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting," *IEEE Trans. Inf. Theory*, vol. 58, no. 5, pp. 3250–3265, May 2012, doi: 10.1109/TIT.2011.2182033.
- [21] G. C. Cawley and N. L. C. Talbot, "On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079–2107, 2010.
- [22] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, Jul. 2009, doi: 10.1016/j.ipm.2009.03.002.
- [23] R. R. Bouckaert and E. Frank, "Evaluating the Replicability of Significance Tests for Comparing Learning Algorithms," 2004, pp. 3–12. doi: 10.1007/978-3-540-24775-3\_3.
- [24] C. Nadeau and Y. Bengio, "Inference for the Generalization Error," *Mach. Learn.*, vol. 52, no. 3, pp. 239–281, Sep. 2003, doi: 10.1023/A:1024068626366.
- [25] S. Holm, "A Simple Sequentially Rejective Multiple Test Procedure," *Scandinavian Journal of Statistics*, vol. 6, pp. 65–70, 1979.