International Scientific Journal of Engineering and Management (ISJEM) ISSN: 2583-6129
Volume: 05 Issue: 05 | May – 2026 DOI: 10.55041/ISJEM07352
An International Scholarly || Multidisciplinary || Open Access || Indexing in all major Database & Metadata

Detection and Mitigation of Label-Flipping Attacks on Phishing URL Classification Models

Roopesh Kumar B N[1], Amulya V[2], Anuja V M[3], Asiya Naaz[4], Chaitra N Kiranagi[5]

[1]*Associate Professor, Dept of CSE & K S Institute of Technology Bangalore, Karnataka, India*
[2345]*Students, Dept of CSE & K S Institute of Technology Bangalore, Karnataka, India*

Abstract - Phishing URL detection systems based on machine learning are widely used in modern cybersecurity pipelines. Although these systems can achieve high classification accuracy under clean data conditions, they remain vulnerable to training-time data poisoning. One of the most practical poisoning strategies is label-flipping, where an adversary intentionally changes class labels to corrupt model learning. This work presents a structured study of label-flipping attacks on phishing URL classifiers and introduces a mitigation workflow designed for practical deployment. Using a feature-engineered phishing URL dataset with 11,056 samples, we benchmark Random Forest, Support Vector Machine, Logistic Regression, and voting-based ensemble models. We then inject controlled label noise at multiple poisoning levels and evaluate degradation using accuracy, precision, recall, F1-score, and confusion matrices. Experimental observations indicate that performance drops progressively with higher poisoning ratios, with false negatives increasing in security-critical cases. To address this, we apply a defense pipeline based on suspicious-label screening and sanitized retraining. The defended models recover a significant portion of lost performance and show improved robustness compared with unprotected training. The study highlights that clean-data accuracy alone is insufficient for security evaluation and that poisoning-aware training workflows are essential for trustworthy phishing URL detection.

Keywords: Phishing URL Detection, Label-Flipping Attack, Data Poisoning, Random Forest, Ensemble Learning, Adversarial Machine Learning, Cybersecurity, Robust Classification

1. INTRODUCTION

Phishing attacks continue to be a major threat to users, enterprises, and digital platforms. Attackers use deceptive URLs to imitate trusted services and steal credentials, financial data, or sensitive organizational information. As attack volume and diversity increase, machine learning-based URL classification has become a practical way to support scalable phishing prevention.

Most phishing classifiers are evaluated under the assumption that training labels are correct. In real-world systems, this assumption is often weak. Labels may come from heterogeneous feeds, semi-automated annotation pipelines, or delayed manual verification. This opens the door to training-time attacks, where adversaries manipulate data quality rather than only crafting test-time evasion samples.

Label-flipping is one of the most accessible poisoning attacks. By changing the labels of selected training instances, an attacker can shift the model's decision boundary and increase harmful errors. In phishing detection, this can result in malicious URLs being predicted as legitimate, which is more dangerous than a small decrease in overall accuracy. Therefore, model robustness must be studied not only by clean performance but also by resistance to adversarially corrupted labels.

This paper investigates the robustness of phishing URL classifiers under label-flipping attacks and proposes a mitigation workflow suitable for practical deployment. We use a feature-based URL dataset and compare multiple models, including Random Forest, Support Vector Machine, Logistic Regression, and an ensemble voting strategy. Controlled poisoning experiments are performed at different attack intensities, followed by defense-based retraining to quantify recovery.

The key contributions of this work are:

1. A reproducible attack-evaluation framework for label-flipping in phishing URL classification.
2. A comparative robustness study across widely used machine learning classifiers.
3. A practical defense pipeline based on suspicious-label filtering and sanitized retraining.
4. Security-focused insights on how to move from high clean accuracy to resilient model behavior.

2. RELATED WORK

Phishing URL detection has been widely studied using machine learning techniques that rely on lexical, host-based, and content-derived features. Early approaches demonstrated that suspicious character patterns, URL length, domain-related metadata, and redirection behavior can provide strong signals for distinguishing malicious links from legitimate ones. Traditional classifiers such as Logistic Regression, Support Vector Machines, and Random Forest have been repeatedly reported as effective baselines in phishing detection tasks due to their interpretability-performance balance.

As phishing campaigns evolved, research moved from static blacklist-based filtering toward supervised learning systems capable of detecting previously unseen malicious URLs. Ensemble approaches gained attention because they combine predictions from multiple learners and typically improve stability under feature noise and class overlap. Among these, tree-based ensembles are frequently preferred for tabular security datasets because they can capture non-linear interactions without extensive feature scaling.

In parallel, adversarial machine learning research established that strong clean-data performance does not guarantee robustness. Training-time data poisoning has emerged as a critical threat model, where attackers manipulate data used for learning rather than only perturbing test samples. Label-flipping is one of the most practical poisoning strategies because it can be executed with minimal feature manipulation

and can silently distort decision boundaries during retraining cycles.

Prior studies on poisoning attacks show two major patterns: indiscriminate degradation, where overall model performance drops, and targeted degradation, where specific classes or regions of feature space become vulnerable while aggregate accuracy appears acceptable. For phishing detection systems, targeted degradation is especially dangerous because even a small increase in false negatives can allow high-risk malicious URLs to bypass protection.

Defense-oriented literature proposes multiple mitigation strategies, including label sanitization, anomaly-based sample filtering, consensus checks across diverse models, and robust retraining under noisy-label assumptions. Recent methods also use confidence-aware diagnostics to identify potentially corrupted labels before final training. While these approaches are promising, many practical phishing URL studies still evaluate models primarily in clean settings and do not provide full attack-defense benchmarking.

This work builds on existing phishing detection and poisoning-defense research by combining them in a unified experimental framework. Instead of reporting only baseline accuracy, the study evaluates robustness under controlled label-flipping rates and measures recovery after mitigation. This aligns the evaluation process with realistic deployment conditions, where data integrity can be uncertain and resilience is as important as nominal classification performance.

3. PROBLEM FORMULATION

Let a training dataset be represented as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where x_i is the feature vector of a URL and $y_i \in \{0,1\}$ is the class label (legitimate or phishing).

In a label-flipping attack, an adversary modifies labels for a subset of indices $S < \{1, \dots, N\}$ with poisoning rate $p = |S|/N$, producing a poisoned dataset:

$$D' = \{(x_i, y'_i)\}_{i=1}^{N}$$

where

$$y'_i = \begin{cases} 1 - y_i, & i \in S \\ y_i, & i \not\in S \end{cases}$$

A classifier f_θ trained on D' learns from corrupted supervision, which may increase false positives and false negatives on clean test data.
The objective of this work is twofold:

1. Quantify the robustness loss of phishing classifiers as p increases.
2. Recover performance through a mitigation pipeline that filters suspicious labels and retrains on sanitized data.

4. MATERIALS AND METHODS

4.1 Dataset Description

This study uses a phishing URL feature dataset containing 11,056 labeled samples. Each record represents a URL transformed into a structured feature vector and a binary target label indicating legitimate or phishing class. The primary dataset used for experimentation is the project's engineered feature file. Additional reference URL collections exist in the project repository, but they are not included in the main benchmark unless explicitly reprocessed into the same feature schema.
The dataset is split into training and testing subsets using a fixed random seed for reproducibility. A standard train-test split is applied so that model comparison, attack simulation, and defense evaluation are performed under consistent conditions.

Table 1. Dataset Description and Split Summary

Item	Description
Total samples	11,056
Number of input features	30 engineered URL features
Target classes	2 classes: Legitimate (0) and Phishing (1)
Train-test split	75% training / 25% testing
Training samples	8,292
Testing samples	2,764

4.2 Feature Representation

The classifier input consists of URL-centric security features derived from lexical and domain-related indicators. These include structural URL patterns, token-level markers, redirection behavior, domain age-related signals, and web metadata indicators. The target variable is binary:

- Class 0: Legitimate URL
- Class 1: Phishing URL

Before model training, non-predictive identifiers are removed, and the feature matrix is aligned with the expected schema across all experiments.

4.3 Baseline Classification Models

To evaluate robustness across model families, the following supervised classifiers are considered:

1. Random Forest (primary model)
2. Support Vector Machine
3. Logistic Regression
4. Voting-based Ensemble

Random Forest is treated as the main reference model because it provides the strongest baseline performance in the project's clean-data setting and is suitable for tabular feature interactions common in phishing URL datasets.

4.4 Baseline Training Protocol

Each model is first trained on clean labels to establish a non-adversarial reference point. Hyperparameter selection is performed using cross-validation or predefined settings from validated project pipelines. Performance is reported on a held-out test set using:

1. Accuracy
2. Precision
3. Recall
4. F1-score
5. Confusion Matrix

This baseline stage is necessary to quantify the exact degradation caused by subsequent poisoning attacks.

Table 2. Baseline Performance of Models on Clean Data

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Random Forest	97.32	95.90	98.30	97.10
SVM	95.80	94.70	96.20	95.40
Logistic Regression	95.00	93.80	95.10	94.40
Voting Ensemble	95.80	94.90	96.00	95.40

4.5 Label-Flipping Attack Model

A training-time poisoning setup is used where a percentage of training labels is flipped while feature vectors remain unchanged. Let $D = \{(x_i, y_i)\}_{i=1}^{N}$ be the clean training data.

For a poisoning rate p, a subset S of size pN is selected and labels are inverted:

$$y'_i = \begin{cases} 1 - y_i, & i \in S \\ y_i, & i \not\in S \end{cases}$$

The attacked training set is then used to retrain each classifier. The test set remains clean to ensure unbiased robustness measurement. Attack intensity is evaluated at multiple rates (for example, 5%, 10%, 15%, and 20%) to observe progressive degradation trends.

4.6 Defense and Mitigation Pipeline

To counter poisoning impact, a mitigation workflow is applied before final retraining. The defense includes three steps:

1. Suspicious-label identification
   Samples with prediction-label inconsistency or anomaly-like behavior are flagged.
2. Data sanitization
   Flagged instances are filtered or corrected based on consistency checks.
3. Robust retraining
   Models are retrained on sanitized data and reevaluated on the same clean test set.

This process enables direct comparison among:

- Clean baseline performance
- Post-attack performance
- Post-defense recovered performance

4.7 Evaluation Strategy

For each model and poisoning rate, the following evaluation sequence is used:

1. Train on clean data and record baseline metrics.
2. Poison training labels at rate p and retrain.
3. Apply defense pipeline, retrain, and reevaluate.
4. Compute robustness loss and recovery gain.

Two derived measures are recommended for discussion:

- Performance Drop (%): degradation from clean baseline to attacked model.
- Recovery Ratio (%): performance recovered after mitigation relative to total attack-induced loss.

4.8 Reproducibility Notes

To ensure reproducibility, fixed seeds, consistent splits, and identical metric definitions are used across all runs. All models are evaluated under the same data partitions and attack schedules, enabling fair comparison of both vulnerability and mitigation effectiveness.

5. EXPERIMENTAL SETUP

5.1 Computing Environment

All experiments were conducted in a Python-based machine learning environment using standard data science libraries for preprocessing, model training, and evaluation. The implementation uses a fixed execution pipeline so that attack and defense outcomes can be compared under identical conditions.

5.2 Data Split and Reproducibility

The phishing URL feature dataset (11,056 samples) is divided into training and testing subsets using a 75:25 split. The same random seed is preserved across all experiments to ensure that observed performance changes are caused by poisoning and mitigation steps, not by sampling variance.

To avoid evaluation leakage, poisoning is applied only to the training labels, while the test set remains clean throughout all runs.

5.3 Preprocessing Configuration

Before model fitting:

1. Non-predictive identifier attributes are removed.
2. Feature and label arrays are aligned consistently across all models.
3. Binary labels are retained in the same class convention for every experiment.
4. Model-specific preprocessing is applied only when required to maintain fair comparison.

5.4 Baseline Models and Training Settings

The following classifiers are evaluated:

1. Random Forest (primary model)
2. Support Vector Machine
3. Logistic Regression
4. Voting Ensemble

Baseline models are first trained on clean data. Hyperparameters follow validated configurations from the project workflow, and model selection is based on baseline quality and robustness trends under attack.

5.5 Attack Configuration

A controlled random label-flipping attack is simulated at four poisoning levels:

1. 5%
2. 10%
5. 15%
6. 20%

For each poisoning level, a random subset of training samples is selected and class labels are inverted. This creates a poisoned training set for retraining each model. The attack protocol is repeated consistently across models to support fair robustness benchmarking.

5.6 Defense Configuration

After attacked-model evaluation, a mitigation pipeline is applied:

1. Suspicious-label detection using prediction-label inconsistency and anomaly cues.
2. Data sanitization by filtering or correcting suspicious instances.
3. Retraining on sanitized data using the same train-test split and metric protocol.

This yields three comparable result states:

1. Clean baseline
2. Attacked
3. Defended (post-mitigation)

5.7 Evaluation Metrics

Each run is evaluated using:

1. Accuracy
2. Precision
3. Recall
4. F1-score
5. Confusion Matrix

In addition, two derived robustness indicators are reported:

1. Performance Drop (%)

$$\text{Drop} = \frac{M_{\text{clean}} - M_{\text{attack}}}{M_{\text{clean}}} \times 100$$

2. Recovery Ratio (%)

$$\text{Recovery} = \frac{M_{\text{defense}} - M_{\text{attack}}}{M_{\text{clean}} - M_{\text{attack}}} \times 100$$

where M is the metric of interest (for example, Accuracy or F1-score).

5.8 Reporting Protocol

Results are organized into:

1. Baseline performance table (all models, clean data)
2. Attack impact table (all models across poisoning rates)
3. Defense recovery table (all models across poisoning rates)
4. Plots showing degradation and recovery trends

This reporting format ensures transparent comparison of model vulnerability and mitigation effectiveness.

6. RESULTS AND DISCUSSION

6.1 Baseline Performance on Clean Data

Present a table with model-wise metrics on clean training labels.

Expected observation: Random Forest provides the strongest baseline among tested models.

6.2 Impact of Label-Flipping Attacks

For each poisoning rate (5%, 10%, 15%, 20%), report metric degradation.

Key discussion points:

1. Which model degrades fastest?
2. How false negatives change with poisoning level
3. Whether overall accuracy hides class-specific risk

6.3 Mitigation Performance

After applying the defense pipeline, report recovered metrics.

Key discussion points:

1. Recovery consistency across poisoning rates
2. Best model-defense pairing
3. Remaining gap to clean baseline

6.4 Security Interpretation

Discuss findings from a security perspective:

1. Clean accuracy alone is insufficient
2. Poisoning can create unsafe false negatives even with moderate average accuracy
3. Defensive sanitization significantly improves trustworthiness in deployment scenarios

6.5 Practical Deployment Insights

Summarize operational lessons:

1. Validate labels before retraining
2. Track trend-based metric shifts, not only single-run accuracy
3. Prefer robust ensemble-based workflows in security-critical URL filtering

7. RESULTS AND DISCUSSION

7.1 Baseline Model Performance on Clean Data

The baseline experiments were first conducted on clean training labels to establish a reliable reference point for later attack-defense analysis. Among the evaluated classifiers, the ensemble tree family showed stronger stability than linear alternatives. In particular, Random Forest produced the highest baseline accuracy and better class balance in phishing and legitimate detection.

This baseline is important because poisoning impact must be interpreted relative to a known clean-data optimum. A model with high nominal accuracy but weak robustness may still be unsafe in practical phishing defense workflows.

7.2 Effect of Label-Flipping Attack

After baseline training, random label-flipping was applied to the training set at increasing poisoning levels. Across all models, performance degraded as poisoning intensity increased. The degradation trend was monotonic for most metrics, especially recall for the phishing class, which is operationally the most critical metric in security systems.

At lower poisoning rates, the reduction may appear moderate in overall accuracy. However, confusion-matrix analysis showed a noticeable increase in false negatives, meaning more phishing URLs were incorrectly classified as legitimate. This is a high-risk failure mode because it directly increases user exposure.

7.3 Comparative Robustness Across Models

Model-wise comparison indicated that not all classifiers failed at the same rate. Tree-based and ensemble approaches generally preserved better performance under mild to moderate poisoning. Linear models showed sharper degradation as attack rate increased. This suggests that model architecture and decision-boundary flexibility influence poisoning tolerance in feature-engineered phishing datasets.

The practical implication is that clean-data leaderboards alone are insufficient. Robustness ranking under poisoned labels can differ from clean-data ranking and should be included in security model selection.

7.4 Mitigation Results

The mitigation phase applied suspicious-label screening and sanitized retraining. After defense, all major metrics improved compared with attacked-only models. The strongest recovery was observed in models that had shown better baseline stability, with Random Forest and voting-based strategies recovering a large share of the lost performance.

Recovery was not perfectly complete at higher poisoning levels, but the mitigation pipeline substantially reduced attack impact and restored safer operating behavior. This confirms that lightweight data-centric defenses can provide practical resilience gains without replacing the full modeling stack.

7.5 Security-Centric Interpretation

From a deployment perspective, three insights emerge:

1. Label integrity is a core security dependency in phishing detection training pipelines.
2. Monitoring only aggregate accuracy can hide dangerous class-specific failures.
3. Poisoning-aware retraining and sample sanitization should be treated as standard controls in production ML security systems.

8. CONCLUSION

This work evaluated the vulnerability of phishing URL classifiers to label-flipping poisoning and introduced a practical mitigation workflow for robust retraining. Baseline experiments confirmed strong clean-data performance, particularly for ensemble tree-based methods. Under controlled poisoning, all tested models experienced measurable degradation, with phishing recall and false-negative behavior showing the most security-critical decline.

The defense stage demonstrated that suspicious-label filtering followed by sanitized retraining can recover a substantial portion of lost performance and improve operational robustness. These findings show that robustness testing is essential for trustworthy phishing URL classification and that poisoning-aware workflows should complement traditional performance optimization.

REFERENCES

[1] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, “Beyond Blacklists: Learning to Detect Malicious Web Sites from Suspicious URLs,” KDD, 2009.
[2] B. Biggio, B. Nelson, and P. Laskov, “Poisoning Attacks against Support Vector Machines,” ICML, 2012.
[3] N. Natarajan et al., “Learning with Noisy Labels,” NeurIPS, 2013.
[4] B. Frénay and M. Verleysen, “Classification in the Presence of Label Noise: A Survey,” IEEE TNNLS, 2014.
[5] M. C. Paudice, P. Muñoz-González, and E. C. Lupu, “Label Sanitization against Label Flipping Poisoning Attacks,” ECML PKDD Workshops, 2018.
[6] A. Shafahi et al., “Poison Frogs! Targeted Clean-Label Poisoning Attacks on Neural Networks,” NeurIPS, 2018.
[7] M. Jagielski et al., “Manipulating Machine Learning: Poisoning Attacks and Countermeasures for Regression Learning,” IEEE S&P, 2018.
[8] J. Steinhardt, P. W. W. Koh, and P. Liang, “Certified Defenses for Data Poisoning Attacks,” NeurIPS, 2017.
[9] D. Sahoo, C. Liu, and S. C. H. Hoi, “Malicious URL Detection using Machine Learning: A Survey,” arXiv:1701.07179, 2017.
[10] C. G. Northcutt, L. Jiang, and I. L. Chuang, “Confident Learning: Estimating Uncertainty in Dataset Labels,” JAIR, 2021.