



Article

The Fragility of Phishing Detection Models: Evidence from Cross-Corpus Transfer, Prevalence Shift, Artifact Learning, and Evasion Risk

Istiaque Bhuiyan ¹ and Tanvir Bhuiyan ^{2,*}

¹ Department of Computer Science, La Trobe University, Kingsbury Drive, Bundoora, VIC 3086, Australia; 22449628@students.latrobe.edu.au

² Murdoch Business School, Murdoch University, 90 South St., Murdoch, WA 6150, Australia

* Correspondence: tanvir.bhuiyan@murdoch.edu.au

Abstract

Phishing detection models often report strong benchmark performance, yet their reliability under realistic deployment conditions remains uncertain. This study examines this problem by investigating three failure modes of cross-dataset phishing email detection: corpus generalization failure, asymmetric prevalence-shift failure, and artifact-driven spurious learning. Using six public email corpora, CEAS_08, Enron, Ling, Nazario, Nigerian Fraud, and SpamAssassin, the study evaluates Term Frequency (TF) and Inverse Document Frequency (IDF)-based Logistic Regression and Linear Support Vector Classifier (SVC) models across pooled baseline testing, single-corpus cross-dataset transfer, leave-one-corpus-out pooled training, prevalence-shift simulation, training prevalence manipulation, dataset-identification analysis, top-feature inspection, artifact-removal ablation, and targeted feature-sensitivity masking. The findings show that single-corpus models are unstable under cross-dataset transfer, with F1-scores varying substantially across source–target combinations. In contrast, leave-one-corpus-out pooled training improves robustness, with Logistic Regression achieving sustained F1-scores between 0.8201 and 0.8994, and Linear SVC achieving F1-scores between 0.7607 and 0.8910 across unseen corpora. Prevalence-shift experiments reveal that failure is asymmetric and threshold-dependent. High-prevalence-trained models maintain high recall under fixed thresholds but suffer sharp recall degradation when operational alert-budget constraints are imposed. Conversely, low-prevalence-trained models become overly conservative in high-threat environments, producing high precision but substantially lower recall and poorer calibration. Artifact analyses further show that source corpus identity is highly learnable, with dataset-identification accuracy reaching 0.9722 for Logistic Regression and 0.9806 for Linear SVC. Top-feature and masking analyses indicate that models rely partly on corpus markers, date tokens, URL/domain terms, headers, and other artifact-like features rather than only general phishing indicators. The study contributes a deployment-aware and adversary-aware evaluation framework for phishing detection. It shows that benchmark accuracy alone is insufficient for assessing real-world robustness and that reliable phishing detection requires cross-corpus validation, prevalence-aware thresholding, and systematic testing for artifact-driven spurious learning.



Academic Editors: Min Chen,
Giancarlo Fortino and Wenjing Xiao

Received: 10 May 2026

Revised: 23 June 2026

Accepted: 26 June 2026

Published: 29 June 2026

Copyright: © 2026 by the authors.
Licensee MDPI, Basel, Switzerland.
This article is an open access article
distributed under the terms and
conditions of the [Creative Commons
Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: phishing detection; cross-dataset generalisation; prevalence shift; artifact learning; machine learning; email security; adversarial robustness; evasion risk

1. Introduction

1.1. Background and Research Problem

Phishing remains one of the most persistent and damaging forms of cyber-enabled deception. By impersonating trusted individuals, organizations, platforms, or institutions, phishing emails attempt to manipulate users into disclosing credentials, clicking malicious links, downloading harmful attachments, or performing unauthorized actions [1–4]. Because phishing attacks exploit both technical weaknesses and human decision-making, machine learning-based phishing detection has become an important area of cybersecurity research. Email-based phishing detection models commonly use textual and metadata-based features from subject lines, message bodies, sender information, URLs, and formatting patterns to classify emails as phishing or legitimate [1,5]. Despite strong progress, a major limitation remains in how phishing detection models are evaluated. Many studies rely on random train–test splits within a single dataset or pooled benchmark dataset. While this approach can show whether a model performs well on held-out samples from the same overall distribution, it may overstate real-world robustness. In operational environments, a model trained on one phishing corpus may be deployed in a different organizational, temporal, linguistic, or threat context. Therefore, high benchmark performance does not necessarily mean that the model will generalize reliably across corpora [6–8]. This issue is particularly important because public phishing datasets are not uniform. They are often collected from different sources, time periods, spam traps, repositories, organizational contexts, and preprocessing pipelines. As a result, each corpus may contain different vocabulary patterns, class distributions, metadata fields, formatting styles, and source-specific artifacts. For example, some email corpora include sender, receiver, date, subject, body, label, and URLs, while others contain only subject, body, and label. Some sources may also be phishing-only collections, which can support positive-class training but cannot support complete binary testing because they lack legitimate emails. These differences mean that cross-dataset phishing detection should be understood as a distribution-shift problem rather than a simple binary classification task [1,9]. A further problem concerns phishing prevalence mismatch. Benchmark datasets are often balanced or contain a relatively high proportion of phishing emails, whereas phishing emails in real organizational email traffic are typically low-prevalence. This mismatch can distort model behavior when training and deployment environments differ. A model trained on high-phishing-prevalence data may maintain high recall under a fixed threshold, but in a low-prevalence environment it may generate too many false alerts. Once operational alert budgets or stricter thresholds are imposed to control false positives, recall may collapse. Conversely, a model trained on low-phishing-prevalence data may become overly conservative and fail to detect enough phishing emails when deployed in a high-threat environment [6,8,10]. Dataset-specific artifacts create another major challenge. If models learn corpus-specific terms, date patterns, email headers, formatting conventions, mailing-list tokens, or anonymization markers, they may perform well in benchmark settings while failing to capture general phishing behavior. In such cases, models may partly learn “which dataset an email came from” rather than “whether the email is phishing.” This creates spurious correlations that can inflate in-dataset performance but weaken cross-dataset generalization. It also introduces a security concern, because artifact-sensitive models may be vulnerable when attackers alter wording, formatting, headers, or other surface-level cues [3,9]. Accordingly, this study investigates cross-dataset phishing detection through three connected failure modes: corpus generalization failure, asymmetric prevalence-shift failure, and artifact-driven spurious learning. The central research problem is that phishing detection models are often evaluated under conditions that do not fully represent real-world deployment, where corpora differ, phishing prevalence shifts, and attackers may manipulate surface-level cues. This

study therefore asks whether pooled multi-corpus training improves robustness, whether prevalence mismatch produces asymmetric recall and calibration failures, and whether dataset-specific artifacts create exploitable weaknesses in phishing detection models.

The central research question is:

How do phishing detection models fail under cross-dataset deployment conditions, and can pooled multi-corpus training, prevalence-aware evaluation, and artifact analysis reveal the limits of benchmark-based phishing detection?

1.2. Research Context

This study examines phishing detection in the context of cross-corpus email classification. The dataset used in this study consists of multiple public email corpora, including CEAS_08, Enron, Ling, Nazario, Nigerian Fraud, and SpamAssassin. These corpora differ in size, class structure, available fields, and source context. Some corpora contain both legitimate and phishing emails, while others are phishing-only sources. This makes the dataset suitable for studying whether phishing detection models generalize across different email sources rather than simply performing well within one dataset [11,12]. The research context is therefore different from standard pooled benchmark evaluation. In a typical random split, training and testing samples may share the same corpus-specific patterns. This can make model performance appear stronger than it would be under true cross-dataset deployment. By contrast, this study evaluates models using single-corpus transfer and leave-one-corpus-out pooled training, where the test corpus is unseen during training. This allows the study to examine whether models trained on one or more corpora can detect phishing emails in a new corpus with different linguistic and structural properties [12,13]. The study also considers phishing detection as an operational deployment problem rather than only a classification problem. In real email systems, phishing prevalence is usually much lower than in benchmark datasets. Therefore, evaluation must consider not only F1-score, accuracy, and ROC-AUC, but also false-positive rates, false-negative rates, recall under alert-budget constraints, and the number of alerts generated per 10,000 emails. These operational metrics are important because even a technically accurate model may be impractical if it creates too many alerts or misses phishing emails under stricter thresholds [14]. Finally, the study investigates dataset artifacts as a source of spurious learning. Corpus-specific tokens, date patterns, header fields, URL/domain markers, and formatting cues may allow models to identify the source dataset rather than general phishing behavior. This research context is important because it connects benchmark weakness to deployment vulnerability. If models rely on dataset-specific shortcuts, then high benchmark performance may not translate into robust real-world protection [13,15]. Therefore, the study frames phishing detection as a problem of cross-corpus generalization, prevalence-aware deployment, and artifact-resistant modeling.

1.3. Research Objectives

Based on the theoretical arguments developed in Section 2, this study pursues four main research objectives:

RO1: To evaluate whether pooled multi-corpus training improves cross-dataset phishing detection compared with single-corpus training.

RO2: To examine whether high-phishing-prevalence training causes recall degradation in low-prevalence deployment environments when operational alert-budget or stricter threshold constraints are applied.

RO3: To assess whether models trained under low-phishing-prevalence conditions become overly conservative when deployed in high-threat environments, resulting in missed phishing emails, degraded recall, and poorer calibration.

RO4: To investigate whether dataset-specific artifacts, including corpus-specific tokens, formatting patterns, metadata fields, and source markers, create spurious correlations that weaken cross-dataset generalization and create potential evasion pathways.

To achieve these objectives, the study develops and evaluates Term Frequency (TF) and Inverse Document Frequency (IDF)-based Logistic Regression and Linear Support Vector Classifier (SVC) phishing detection models across multiple experimental settings. These include pooled baseline testing, single-corpus cross-dataset transfer, leave-one-corpus-out pooled training, prevalence-shift simulation, training prevalence manipulation, dataset-identification experiments, top-feature artifact analysis, artifact-removal ablation, and targeted feature-sensitivity masking.

1.4. Summary of Findings

The empirical analysis yields four main findings. First, the results show that single-corpus phishing models are unstable under cross-dataset transfer. Models trained on one corpus and tested on another produced large variations in F1, MCC, false-positive rate, and false-negative rate. Some source–target combinations performed reasonably well, while others showed major degradation. This confirms that strong performance on one corpus does not guarantee reliable generalization to another corpus.

Second, pooled multi-corpus training improved cross-dataset robustness. In the leave-one-corpus-out experiments, pooled-minus-test training produced stronger and more sustained F1 performance than the best single-corpus transfer baselines. This pattern was observed for both Logistic Regression TF-IDF and Linear SVC TF-IDF. The results support the argument that exposure to multiple corpora can reduce dependence on a single source distribution and improve generalization to unseen corpora.

Third, the prevalence-shift experiments showed that model failure is asymmetric and threshold-dependent. High-prevalence training deployed in low-prevalence environments did not cause recall collapse under fixed-threshold evaluation. However, when realistic alert-budget constraints were applied, recall dropped sharply because the model was allowed to flag only a small proportion of emails. In the opposite direction, low-prevalence training did not cause precision collapse. Instead, models became conservative: precision remained high, but recall and F1 degraded because many phishing emails were missed in high-threat deployment environments.

Fourth, the artifact-analysis results showed that dataset-specific artifacts are highly learnable and influence model behavior. Dataset-identification models could predict the source corpus with high accuracy, indicating that the corpora contain strong source-specific signals. Top-feature analysis revealed that models used corpus markers, date tokens, URL/domain terms, and other artifact-like features. Artifact-removal and targeted feature-sensitivity analysis further showed that removing or masking high-weight learned features reduced detection performance and improved evidence for artifact reliance. These findings support the claim that benchmark models can learn spurious correlations that weaken cross-dataset robustness and create a plausible pathway from benchmark weakness to deployment vulnerability.

1.5. Contributions and Paper Structure

This study makes several contributions to the literature on machine learning-based phishing detection and cybersecurity model evaluation. First, it reframes phishing detection as a deployment robustness problem rather than only a benchmark classification problem. While prior studies often report strong model performance using random train–test splits, this study shows that such evaluation can overstate real-world reliability when training and deployment corpora differ. By examining single-corpus transfer and leave-one-corpus-out

pooled training across CEAS_08, Enron, Ling, Nazario, Nigerian Fraud, and SpamAssassin, the study provides evidence that phishing detection models must be assessed under cross-corpus conditions that better reflect operational deployment environments.

Second, the study contributes by identifying corpus generalization failure as a major limitation of single-source phishing detection models. The findings show that models trained on one corpus often perform inconsistently when tested on another unseen corpus, indicating that strong performance within one dataset does not necessarily translate into generalizable phishing detection. In contrast, pooled multi-corpus training improves robustness by exposing models to more diverse phishing and legitimate email patterns. This contribution is important because it provides empirical support for cross-dataset validation as a necessary step in evaluating phishing detection systems.

Third, the study contributes to prevalence-aware cybersecurity evaluation by demonstrating that model failure under prevalence shift is asymmetric and threshold-dependent. High-prevalence training does not automatically cause recall collapse under fixed-threshold evaluation. However, when low-prevalence deployment is combined with realistic alert-budget constraints, recall declines sharply because only a small proportion of emails can be flagged for review. Conversely, low-prevalence-trained models do not necessarily suffer precision collapse in high-threat environments; rather, they become overly conservative, producing lower recall, more missed phishing emails, and poorer calibration. This distinction clarifies how phishing prevalence mismatch affects operational performance and why fixed-threshold benchmark metrics are insufficient for deployment evaluation.

Fourth, the study contributes by linking dataset-specific artifacts to spurious learning and possible evasion risk. Through dataset-identification experiments, top-feature inspection, artifact-removal ablation, and targeted feature-sensitivity masking, the study shows that phishing detection models may partly learn corpus markers, date tokens, header fields, URL/domain terms, formatting patterns, and other artifact-like features. This indicates that benchmark performance may partly reflect source recognition rather than genuine phishing detection. The study therefore advances an artifact-aware evaluation framework that connects model generalization weakness with potential adversarial vulnerability.

Finally, the study offers practical implications for cybersecurity researchers and practitioners. The findings suggest that reliable phishing detection requires more than high benchmark accuracy. Researchers should report cross-corpus validation, prevalence-shift performance, alert-budget sensitivity, calibration behavior, and artifact-dependence diagnostics. Practitioners should also adjust thresholds according to deployment prevalence and operational alert capacity, rather than relying solely on default model thresholds.

The remainder of the paper is structured as follows. Section 2 presents the theoretical background and develops the hypotheses related to cross-dataset generalization, prevalence-shift failure, conservative under-detection, and artifact-driven spurious learning. Section 3 describes the data, preprocessing procedures, model selection, experimental design, prevalence-shift simulations, dataset-identification experiments, artifact-removal ablation, and artifact transformation and feature-sensitivity analysis. Section 4 reports the empirical results, including pooled baseline performance, single-corpus transfer results, leave-one-corpus-out pooled training, prevalence-shift outcomes, training prevalence manipulation, and artifact-analysis findings. Section 5 discusses the results in relation to the hypotheses and explains their implications for robust phishing detection. It concludes the paper by summarizing the main findings, outlining practical implications, acknowledging limitations, and suggesting directions for future research.

2. Theoretical Background and Hypotheses Development

2.1. Cross-Dataset Phishing Detection as a Generalization Problem

Phishing detection has commonly been framed as a supervised binary classification task in which models distinguish phishing emails from legitimate emails using textual, structural, or metadata-based features [1–3]. In email phishing detection, machine learning models typically learn lexical and statistical patterns from subject lines, body text, URLs, sender information, formatting features, and other available metadata [1,5]. These models often achieve strong performance when trained and tested on randomly split samples from the same dataset. However, strong within-dataset performance does not necessarily indicate that a model will generalize reliably to emails drawn from a different corpus or deployment environment [6–8]. This distinction is important because phishing datasets are rarely homogeneous. Public phishing corpora are often collected from different sources, time periods, organizational contexts, spam traps, mailing lists, or curated phishing repositories [1,9]. As a result, each corpus may contain different class distributions, vocabulary patterns, formatting conventions, metadata availability, and collection-specific artifacts. For example, some datasets may include sender, receiver, date, subject, body, label, and URLs, while others may contain only subject, body, and label. Some corpora may also be phishing-only collections, which provide useful positive-class examples but cannot support complete binary evaluation because they lack legitimate emails. These differences make cross-dataset phishing detection a distribution-shift problem rather than a simple binary classification exercise [6,9]. From a machine learning perspective, a model trained on one corpus may learn patterns that are predictive within that corpus but unstable outside it [7,10]. This can occur when the model captures dataset-specific cues instead of general phishing indicators. For example, corpus-specific names, repeated date tokens, header structures, mailing-list patterns, URL formats, or anonymization markers may become statistically associated with either phishing or legitimate labels. If these patterns are absent or differently distributed in another dataset, the model may suffer performance degradation. This creates a core problem for phishing detection research: benchmark performance can overstate deployment robustness when evaluation is limited to a single dataset or a random split [6,8]. Cross-dataset evaluation is therefore necessary because real-world phishing detection systems are not deployed in the same controlled conditions under which benchmark datasets are collected. An operational system may be trained on historical phishing corpora but deployed in a new organization, time period, or threat environment. In such settings, the model must recognize generalizable phishing behavior rather than memorizing corpus-specific regularities [2,3]. This study therefore treats cross-dataset phishing detection as a generalization problem and examines how models fail when training and deployment distributions differ. One proposed mitigation is pooled multi-corpus training. By exposing the model to multiple corpora during training, pooled training may reduce dependence on any one dataset's artifacts and increase the diversity of phishing and legitimate examples [6–8]. In theory, a model trained across several corpora should learn broader phishing patterns than a model trained on a single corpus. This should produce more stable F1-scores and lower sensitivity to dataset-specific variation when evaluated on unseen corpora. Accordingly, the first hypothesis is:

H1. *Pooled multi-corpus training outperforms single-corpus models by producing higher sustained F1-scores and lower false-positive rates across unseen phishing email corpora.*

2.2. Prevalence Shift, Alert Budgets, and Recall Collapse

Cross-dataset failure is not caused only by differences in vocabulary or formatting. It can also arise from differences in class prevalence between the training dataset and the deployment environment. Many public phishing datasets are relatively balanced or contain a high proportion of phishing emails because they are designed for model development and evaluation [1,3]. However, real organizational email environments are typically low-prevalence settings in which phishing emails represent a small proportion of total email traffic. This mismatch creates an important deployment challenge because models trained under benchmark-like prevalence conditions may behave differently when applied to operational email streams [6,8]. In a fixed-threshold classification setting, low phishing prevalence alone does not necessarily reduce recall. Recall measures the proportion of true phishing emails correctly detected. Therefore, if the model threshold remains unchanged, the true positive rate may remain relatively stable even when the number of phishing emails in the test environment changes [10]. However, operational deployment rarely depends only on fixed-threshold performance. In practice, organizations must manage alert volume. A model that produces high recall in a low-prevalence environment may also generate an unmanageable number of false positives. A classifier that achieves high recall may still be impractical if it produces too many false positives, because each false alert consumes analyst time and may reduce user trust in the detection system [2,16]. Even a modest false-positive rate can produce thousands of alerts when applied to large volumes of legitimate email traffic. This means that low-prevalence deployment often forces stricter thresholding or alert-budget constraints. Security teams may only investigate the top 1%, 5%, or 10% of highest-risk messages because analyst time and operational resources are limited. Under these constraints, a model trained in a high-phishing-prevalence environment may appear effective under fixed-threshold evaluation but fail when only a small number of emails can be flagged. These operational constraints change the evaluation problem from simple classification accuracy to alert prioritization under limited security resources [3,9]. The model may rank some phishing emails below the alert threshold, resulting in many phishing emails being missed [11,14]. This produces recall collapse not because low prevalence mechanically reduces recall, but because low-prevalence deployment changes the operational thresholding problem [14]. This distinction is theoretically important because it separates statistical evaluation from operational performance. A model may maintain high recall at a fixed threshold while becoming impractical due to alert overload [17]. Conversely, when thresholds are tightened to control false positives, recall may collapse [14,17]. Therefore, the relevant failure mode is the interaction between high-prevalence training, low-prevalence deployment, and alert-budget constraints [11,17]. This study explicitly tests this interaction through prevalence-shift simulation and alert-budget evaluation. Accordingly, the second hypothesis is:

H2. *High-phishing-prevalence training deployed in low-prevalence environments causes recall collapse when operational alert-budget or stricter threshold constraints are applied.*

2.3. Low-Prevalence Training and Conservative Failure in High-Threat Deployment

The opposite prevalence mismatch can also create failure, but through a different mechanism. When a model is trained under low-phishing-prevalence conditions, it is exposed to relatively few phishing examples compared with legitimate examples. As a result, the model may learn a conservative decision boundary that strongly favors the legitimate class [6,7,18,19]. This can produce high precision because the model flags only the most obvious phishing emails, but it can also produce low recall because many phishing messages are not detected [20–22]. This issue becomes serious when the model is deployed

in a high-threat environment where phishing prevalence is much higher than in the training data. In such a setting, a conservative model may continue to avoid false positives, but this comes at the cost of missing many true phishing emails. Therefore, the expected failure is not necessarily precision collapse. In fact, precision may remain high because the model is reluctant to classify emails as phishing. The more likely failure is recall degradation, increased false negatives, and poorer calibration [18,23,24]. This point is important because precision and recall do not fail symmetrically under prevalence shift. When phishing prevalence increases, precision may improve if the model's true-positive and false-positive rates remain stable. However, if the model has been trained under low phishing prevalence, it may become under-sensitive to phishing indicators. The result is an overly cautious detector that performs poorly in high-threat environments because it fails to flag enough phishing emails. In cybersecurity terms, this is particularly dangerous because false negatives represent phishing emails that bypass detection and reach users [20,22,25]. Calibration is also relevant in this setting. A model trained under one class-prevalence condition may produce probability estimates or decision scores that do not align with a different deployment base rate. For probabilistic models such as Logistic Regression, low-prevalence training may lead to underconfident phishing probabilities in high-prevalence settings. This can cause the model to misrepresent the true level of phishing risk and further reduce detection effectiveness when thresholds are applied [18,23,24]. Therefore, this study reframes the third failure mode away from "precision collapse" and toward conservative under-detection. The theoretical expectation is that low-prevalence training produces models that are too cautious when deployed in high-prevalence threat environments. Accordingly, the third hypothesis is:

H3. *Models trained under low-phishing-prevalence environments become overly conservative when deployed in high-threat environments, leading to missed phishing emails, degraded recall, and poorer calibration.*

2.4. Dataset-Specific Artifacts and Spurious Correlations

A further cross-dataset failure mode arises from dataset-specific artifacts. In phishing email corpora, artifacts may include header fields, date patterns, source names, collection-specific vocabulary, formatting conventions, mailing-list tokens, anonymization markers, URL/domain structures, and repeated boilerplate. These features may be statistically predictive within a dataset but may not represent genuine phishing behavior. When models learn these features, they may perform well under benchmark conditions while failing to generalize to new corpora [12,15,26]. This problem is closely related to spurious correlation. A spurious correlation occurs when a feature appears predictive because of the dataset collection process rather than because it reflects the underlying class concept. In phishing detection, a model may associate terms such as dataset names, specific years, email list structures, or source-specific formatting with phishing or legitimacy [3,26]. For example, if a particular corpus contains mostly legitimate organizational emails, a model may learn that corpus-specific terms indicate legitimacy. Conversely, if another corpus contains mostly phishing emails, source-specific tokens from that corpus may become falsely associated with phishing. The risk is that the classifier may partially learn "where the email came from" rather than "whether the email is phishing." This is especially problematic in cross-dataset evaluation because corpus identity is not a stable deployment signal. A real-world phishing email may not contain the same corpus markers as a benchmark phishing email. Similarly, legitimate emails in a new organization may not resemble the legitimate corpus used during training. As a result, artifact reliance can inflate in-dataset performance while weakening cross-dataset robustness [12,15,26]. Dataset artifacts also create a security concern. If model decisions are influenced by surface-level tokens, format-

ting patterns, or header structures, then attackers may be able to manipulate these features. Even if simple surface transformations do not fully evade detection, the existence of artifact sensitivity indicates a plausible vulnerability pathway. Attackers could potentially alter wording, remove suspicious markers, insert benign-looking boilerplate, manipulate formatting, or imitate legitimate corpus patterns to reduce confidence in detection [12]. The interaction between dataset artifacts and adversarial manipulation is important because artifact-sensitive models may rely on features that are easier to modify than the underlying malicious intent of the email. Adversarial machine learning research shows that classifiers can be vulnerable when small, carefully chosen input changes alter the model's prediction while preserving the attacker's goal [27,28]. In text classification, adversarial examples may involve word substitution, character-level changes, paraphrasing, or insertion of distracting content that changes model behavior without necessarily changing the message's meaning [29]. In phishing detection, dataset artifacts such as corpus markers, date tokens, header patterns, URL/domain cues, formatting styles, and repeated boilerplate may therefore become security-relevant if the model treats them as strong phishing indicators. An attacker may not know the exact top-weighted features of a model, but if a detector is sensitive to surface-level cues, then rewriting, formatting changes, benign-looking footer insertion, URL obfuscation, or header manipulation could reduce detection confidence. Therefore, artifact learning should be understood not only as a generalization problem, but also as a potential adversarial-risk pathway.

Therefore, dataset artifacts are not only methodological weaknesses; they may also become exploitable weaknesses in adversarial settings. This study examines artifact reliance through dataset-identification experiments, top-feature analysis, artifact-removal ablation, and targeted masking. If models can accurately predict corpus identity, this indicates that source-specific patterns are highly learnable. If removing or masking artifacts improves cross-dataset generalization or reduces confidence in detection, this suggests that model decisions are partly shaped by spurious features [3,15,26]. Accordingly, the fourth hypothesis is:

H4. *Cross-dataset phishing models learn dataset-specific artifacts, including headers, formatting patterns, boilerplate phrases, and corpus-specific tokens, producing spurious correlations that reduce generalization and create exploitable evasion pathways.*

2.5. Conceptual Framing of the Study

Taken together, the preceding arguments suggest that phishing detection should not be evaluated only through pooled random train–test splits or single-dataset benchmark performance. Such evaluation designs may overstate model robustness by allowing corpus-specific patterns to appear in both training and testing data [12,30]. A more realistic evaluation framework must consider whether models generalize across corpora, whether they remain reliable under prevalence mismatch, and whether they rely on dataset-specific artifacts [14,15].

This study therefore conceptualizes cross-dataset phishing detection failure as three linked failure modes. The first failure mode concerns corpus generalization failure, where single-corpus models perform inconsistently when tested on unseen datasets [11,12]. Pooled multi-corpus training is examined as a mitigation strategy because exposure to multiple corpora may improve robustness and reduce dependence on one source distribution.

The second and third failure modes concern asymmetric prevalence-shift failure. High-prevalence training deployed in low-prevalence environments may not cause recall collapse under fixed thresholds, but it can cause recall collapse when operational alert budgets are imposed. Conversely, low-prevalence training deployed in high-prevalence environments may not cause precision collapse; instead, it can produce conservative models that miss

phishing emails and show degraded recall or calibration [14,24]. These two hypotheses therefore distinguish the direction of prevalence mismatch and the specific metric most likely to fail.

The fourth failure mode concerns artifact-driven spurious learning. If models learn dataset identifiers, header patterns, date tokens, corpus-specific words, or formatting artifacts, then benchmark performance may reflect dataset recognition rather than genuine phishing detection. This creates a pathway from benchmark weakness to deployment vulnerability. In this sense, the paper moves from a traditional performance-evaluation perspective toward an adversary-aware view of phishing detection [12,15].

Overall, the theoretical framework positions robust phishing detection as a problem of cross-corpus generalization, prevalence-aware deployment, and artifact-resistant modeling. This framing supports the central argument of the study: phishing detection models should be evaluated not only by how well they perform on a held-out sample, but also by how and why they fail when the training and deployment environments differ [11,14,30].

3. Data and Methodology

3.1. Research Design

This study adopts a quantitative experimental research design to examine how phishing detection models fail under cross-dataset deployment conditions. The design follows the conceptual framework of three failure modes: corpus generalization failure, asymmetric prevalence-shift failure, and artifact-driven spurious learning. Multi-corpus email data from CEAS_08, Enron, Ling, Nazario, Nigerian Fraud, and SpamAssassin are processed using text and metadata fields, transformed into TF-IDF features, and classified using Logistic Regression and Linear SVC models.

First, the study tests corpus generalization failure by comparing single-corpus cross-dataset transfer with leave-one-corpus-out pooled training. This design evaluates whether pooled multi-corpus training improves robustness when models are tested on unseen email corpora, corresponding to H1.

Second, the study examines asymmetric prevalence-shift failure through two deployment scenarios. For H2, models trained under high phishing prevalence are tested in low-prevalence environments under fixed thresholds and operational alert-budget constraints. For H3, models trained under low phishing prevalence are tested in high-threat environments to assess whether they become conservative, miss phishing emails, and show poorer calibration.

Third, the study investigates artifact-driven spurious learning by testing whether models learn corpus-specific tokens, headers, dates, URLs, and formatting cues instead of genuine phishing indicators. Dataset-identification experiments, top-feature analysis, artifact-removal ablation, and targeted masking are used to evaluate this mechanism, corresponding to H4.

Overall, the research design moves beyond standard benchmark testing by evaluating phishing detection models under realistic deployment conditions involving unseen corpora, changing phishing prevalence, alert-budget limits, and dataset-specific artifacts. Model performance is assessed using F1-score, recall, precision, MCC, false-positive rate, false-negative rate, calibration, alert burden, and robustness under artifact masking. Figure 1 below illustrates the research design framework of the study.

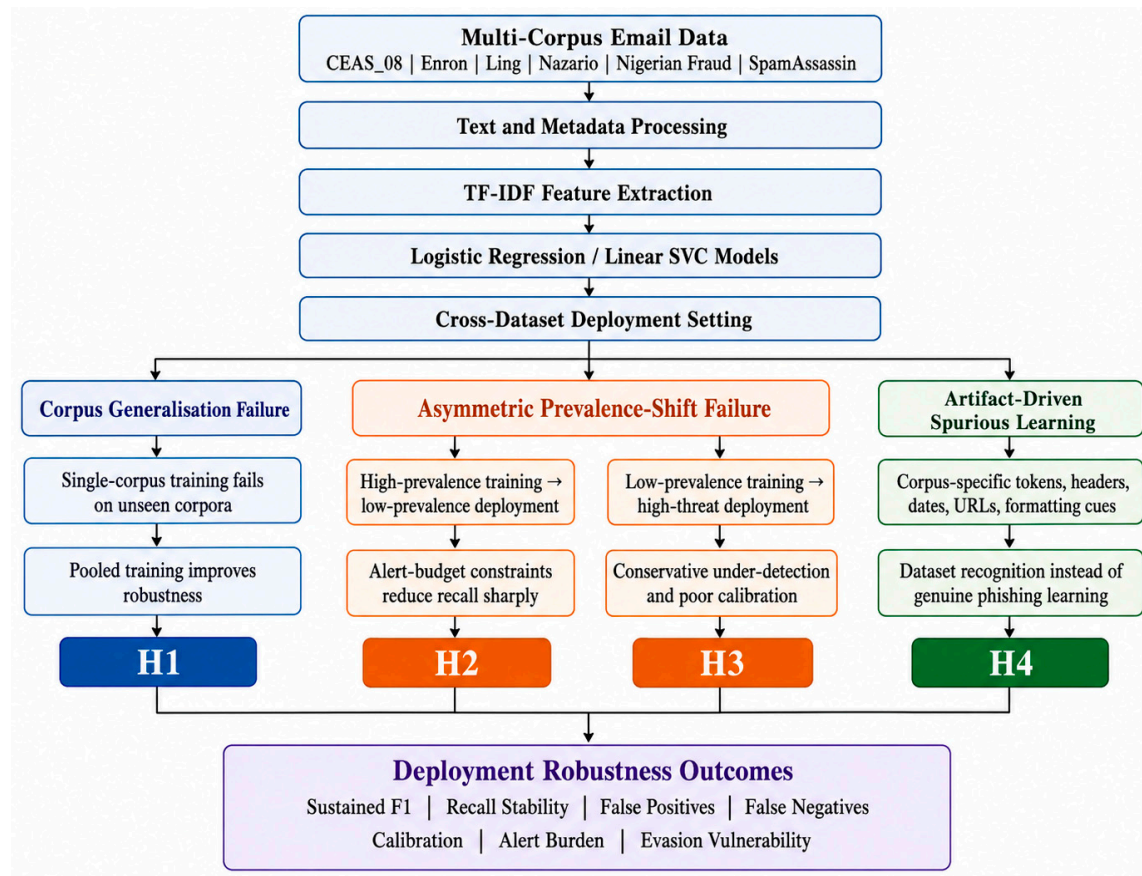


Figure 1. Research design framework.

3.2. Data Description

Six phishing datasets were used in this study. Together, these datasets allowed us to examine the three failure modes of cross-dataset phishing detection. The phishing email dataset (see Table 1 below) was obtained from Kaggle and is available under the CC BY-SA 4.0 license. The dataset was compiled from multiple public email sources, including Enron, Ling, CEAS, Nazario, Nigerian Fraud, and SpamAssassin. It contains both email text and, for some subsets, contextual metadata such as sender, receiver, date, and URLs. The dataset has been associated with recent work on interpretable and robust phishing email detection [1].

Table 1. Phishing email dataset.

Email Source	Number of Emails	Main Available Fields
CEAS_08	39,154	Sender, receiver, date, subject, body, label, URLs
Enron	29,767	Subject, body, label
Ling	2859	Subject, body, label
Nazario	1565	Sender, receiver, date, subject, body, URLs, label
Nigerian Fraud	3332	Sender, receiver, date, subject, body, URLs, label
SpamAssassin	5809	Sender, receiver, date, subject, body, label, URLs
Total	82,486	Combined email dataset

The phishing email dataset contains approximately 82,486 emails, consisting of 42,891 spam/phishing emails and 39,595 legitimate emails. This gives a relatively balanced distribution (Table 2 below) of approximately 52.00% phishing/spam and 48.00% legitimate emails. This dataset was suitable for training and evaluating text-based phishing detection

models because it contains email subject lines, body text, and binary phishing/legitimate labels across multiple sources.

Table 2. Class distribution: Phishing email dataset.

Class	Number of Emails	Percentage
Phishing/spam	42,891	52.00%
Legitimate	39,595	48.00%
Total	82,486	100.00%

Dataset Selection Bias and Generalizability Boundaries

Although the six public corpora provide useful diversity for cross-corpus evaluation, they also introduce important generalizability boundaries. The corpora were collected from different sources, time periods, organizational contexts, spam traps, public repositories, and preprocessing pipelines. As a result, their vocabulary, metadata availability, formatting conventions, phishing prevalence, and label composition may differ from contemporary organizational email streams. Some corpora contain both legitimate and phishing emails, while others are phishing-only sources, meaning that they can support positive-class training but not complete binary deployment testing. In addition, public corpora may contain dated attack templates, anonymization markers, mailing-list patterns, repeated headers, or source-specific formatting artifacts. Therefore, the results should be interpreted as evidence of robustness across selected public corpora rather than as a direct guarantee of performance in all modern-enterprise email environments.

3.3. Data Preprocessing

The email datasets were first standardized into a common structure because the original sources contained different column formats. Column names were normalized by converting them to lowercase and removing leading or trailing spaces. The subject and body fields were extracted where available. If either field was missing, it was replaced with an empty string to prevent processing errors. The subject and body were then combined into a single text field. This combined field was used as the main input for the email classification models. The original class labels were also standardized. Labels such as spam, phishing, malicious, fraud, and 1 were mapped to 1, representing phishing or spam. Labels such as legitimate, ham, benign, and 0 were mapped to 0, representing legitimate emails. The final standardized email data frame contained three core columns: source, text, and label_num. Table 3 illustrates the preprocessing pipeline of the email dataset.

Table 3. Preprocessing pipeline.

Preprocessing Step	Description	Purpose
Column standardization	Converted column names to lowercase and stripped extra spaces	Ensured consistent column handling across datasets
Missing text handling	Replaced missing subject/body fields with empty strings	Prevented text-processing errors
Text construction	Combined subject and body into one text field	Created a unified text input for modeling
Label standardization	Mapped phishing/spam labels to 1 and legitimate labels to 0	Created a consistent binary target variable
Source tracking	Added a source column	Preserved dataset origin for subgroup and cross-source analysis

The email phishing dataset was organized by source corpus. The main usable corpora are described in Table 4. Nazario and Nigerian_Fraud were not used as held-out binary

test corpora because they contain only phishing emails. However, they were included as phishing-positive training sources in the main pooled setting, and an additional sensitivity analysis was conducted to assess whether their inclusion distorted the feature distribution or inflated artifact learning.

Table 4. Dataset classification.

Corpus	Class Structure	Used for Binary Testing	Used for Training
CEAS_08	Legitimate + phishing	Yes	Yes
Enron	Legitimate + phishing	Yes	Yes
Ling	Legitimate + phishing	Yes	Yes
SpamAssassin	Legitimate + phishing	Yes	Yes
Nazario	Phishing only	No	Yes
Nigerian_Fraud	Phishing only	No	Yes

3.4. Model Selection

The full email dataset was combined into one pooled dataset. Two baseline machine learning models were trained using text features. The models were developed using TF-IDF vectorization and linear classifiers. The text input was the combined email subject and body. TF-IDF was used because it provides an interpretable numerical representation of text by weighting terms according to their importance within a document relative to the whole corpus. Specifically, TF measures how often a term appears in a document, while IDF reduces the weight of common terms and increases the weight of rarer, more informative terms. Thus, TF-IDF helps the model identify words that are more meaningful for distinguishing phishing/spam emails from legitimate emails. Both Logistic Regression and Linear SVC used class balancing to reduce bias toward the majority class.

The main experiments used word unigram TF-IDF as the primary feature representation because the study aimed to examine cross-corpus robustness, top-feature behavior, and artifact learning using interpretable textual features. Unigram TF-IDF allowed direct inspection of high-weight tokens, including corpus markers, date terms, URL/domain tokens, and other artifact-like features. However, phishing emails may also contain phrase-level cues, such as “click here”, “verify account”, and “reset password”, as well as character-level patterns associated with spelling variation, URL fragments, and obfuscation.

To address this, an additional robustness check was conducted using two alternative TF-IDF representations: word 1–2-g TF-IDF and character 3–5-g TF-IDF. The same LOCO evaluation design, classifiers, train–test structure, and evaluation metrics were retained, while only the feature representation was changed. The purpose of this check was to assess whether the main findings were dependent on the unigram representation or remained stable when phrase-level and character-level features were included.

3.5. Single-Corpus Cross-Dataset Transfer Experiment

This step was conducted to test H1, which proposes that pooled multi-corpus training improves cross-dataset robustness compared with single-corpus models. As a baseline for H1, this experiment first examined whether a model trained on one corpus could generalize to another unseen corpus. This is important because real phishing detection systems are often trained on one dataset but deployed in another organizational, temporal, or threat environment. Each model was therefore trained on one corpus and tested on a different corpus. For consistency across datasets, the same preprocessing and feature-extraction procedures were applied to all train–test combinations, as reported in Table 5. The 1000-character cap was used so the full cross-corpus TF-IDF experiment could run reliably in this environment.

Table 5. Single-corpus cross-dataset feature extraction.

Component	Setting
Text fields	Subject + Body
Maximum text length	First 1000 characters per email
Feature extraction	TF-IDF
N-grams	Unigram only
Maximum features	5000
Minimum document frequency	10
Sublinear TF scaling	True
Lowercasing	True
Accent stripping	Unicode

The details of the parameters used by the Logistic Regression TF-IDF and Linear SVC TF-IDF are given below in Table 6.

Table 6. Model parameters: Single-corpus cross-dataset transfer experiment.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Classifier	Logistic Regression	Linear SVC
Optimization/solver	liblinear	liblinear-based Linear SVC implementation
Regularization strength	1.0	1.0
Maximum iterations	300	3000
Random state	42	42

3.6. Leave-One-Corpus-Out (LOCO) Pooled-Training Experiment

For each binary corpus, one corpus was held out as the test set. The models were trained on all other available corpora, as shown in Table 7. This experiment was conducted to test H1 by examining whether pooled multi-corpus training improves cross-dataset robustness when the model is evaluated on an unseen corpus. For consistency across datasets, the test preprocessing, feature extraction, 1000-character cap, and model parameters were kept the same as those reported in Tables 5 and 6.

Table 7. Held-out test corpus and corresponding training corpora.

Held-Out Test Corpus	Trained Corpora
CEAS_08	Enron + Ling + Nazario + Nigerian_Fraud + SpamAssassin
Enron	CEAS_08 + Ling + Nazario + Nigerian_Fraud + SpamAssassin
Ling	CEAS_08 + Enron + Nazario + Nigerian_Fraud + SpamAssassin
SpamAssassin	CEAS_08 + Enron + Ling + Nazario + Nigerian_Fraud

Phishing-Only-Corpus Sensitivity Analysis

Nazario and Nigerian_Fraud were treated differently from the binary corpora because they contain phishing emails only and do not include legitimate comparison emails. As a result, they could not be used as held-out binary test corpora for calculating complete classification metrics such as false-positive rate, MCC, or balanced accuracy. However, they were included in the main pooled training setting as phishing-positive sources to increase phishing-language diversity.

To assess whether the inclusion of these phishing-only corpora distorted the feature distribution or introduced source-specific artifacts, a sensitivity analysis was conducted using two training conditions. The first condition included Nazario and Nigerian_Fraud in the pooled training data, matching the main pooled LOCO setup. The second condition excluded Nazario and Nigerian_Fraud from the pooled training data. The same LOCO

evaluation design, binary held-out test corpora, preprocessing pipeline, TF-IDF feature representation, classifiers, and evaluation metrics were retained across both conditions. Logistic Regression TF-IDF and Linear SVC TF-IDF were then compared using mean LOCO F1 and mean LOCO MCC.

In addition, source-identification sensitivity was assessed to determine whether corpus identity remained learnable after removing the phishing-only corpora. Source-identification accuracy was compared between the full six-corpus setting and the reduced setting excluding Nazario and Nigerian_Fraud. This analysis tested whether artifact learning was primarily driven by the phishing-only corpora or reflected broader source-specific differences across the public phishing datasets.

3.7. Prevalence-Shift Simulation

The purpose of this experiment was to simulate a real-world deployment problem and to test H2, which predicts that high-phishing-prevalence training deployed in low-prevalence environments causes recall collapse when operational alert-budget or stricter threshold constraints are applied. In many benchmark datasets, phishing prevalence is high. However, in real organizational email traffic, phishing emails are usually much rarer. If a model trained in a high-phishing-prevalence setting is deployed in a low-prevalence environment, the organization may apply stricter alert thresholds to avoid too many false positives. Therefore, this experiment compared two thresholding conditions. The first was a fixed-threshold setting, where Logistic Regression used a probability threshold of 0.5 and Linear SVC used a decision-score threshold of 0. The second was an operational alert-budget setting, where only the top 1%, 5%, or 10% of highest-risk emails were flagged. The models were trained under a high-phishing-prevalence condition, with a target training phishing prevalence of 70%. The training sample contained 12,000 emails, consisting of approximately 8400 phishing emails and 3600 legitimate emails. To represent different deployment environments, the test sets were resampled at phishing-prevalence levels of 1%, 5%, 10%, 20%, and 50%. These levels represented very-low-prevalence, low-prevalence, moderate-low-prevalence, moderate-prevalence, and high-prevalence benchmark-like deployment settings, respectively. The experiment was repeated across the four binary corpora: CEAS_08, Enron, Ling, and SpamAssassin. The detail model and parameter settings are given below in Table 8.

Table 8. Model parameters: Prevalence-shift simulation.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Classifier	Logistic Regression	Linear SVC
Feature representation	TF-IDF	TF-IDF
N-gram range	Unigram	Unigram
Maximum TF-IDF features	3000	3000
Minimum document frequency	5	5
Sublinear TF scaling	True	True
Text length cap	First 1000 characters	First 1000 characters
Solver/implementation	liblinear	liblinear-based Linear SVC
Max iterations	300	3000
Random state	42	42

3.8. Training Prevalence Manipulation

This experiment was conducted to test H3, which predicts that models trained under low-phishing-prevalence conditions become overly conservative when deployed in high-threat environments. The logic is that if a model sees very few phishing emails during training, it may learn to classify emails conservatively. When deployed in a high-threat environment, this conservative behavior may cause the model to miss many phishing

emails. Therefore, the training data were resampled into five phishing-prevalence settings: 5%, representing very-low-phishing-prevalence training; 10%, representing low-phishing-prevalence training; 30%, representing moderate-phishing-prevalence training; 50%, representing balanced training; and 70%, representing high-phishing-prevalence training. The test set was fixed at 50% phishing prevalence, representing a high-threat deployment environment.

The details of the parameters used by both the models are given below in Table 9. Calibration error was calculated for Logistic Regression because it provides probability outputs. For Linear SVC, calibration error was not reported because standard Linear SVC produces decision scores rather than calibrated probabilities.

Table 9. Model parameters: Training prevalence manipulation.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Classifier	Logistic Regression	Linear SVC
Feature Representation	TF-IDF	TF-IDF
N-gram Range	Unigram	Unigram
Maximum TF-IDF Features	3000	3000
Minimum Document Frequency	5	5
Sublinear TF Scaling	True	True
Text Length Cap	First 800 characters	First 800 characters
Solver/Implementation	liblinear	liblinear-based Linear SVC
Max Iterations	300	3000
Random State	42	42
Training Sample Size	6000 emails	6000 emails
Test Sample Size	4000 emails	4000 emails

3.9. Dataset-Identification Experiment

This experiment was conducted to test H4, which proposes that cross-dataset phishing models learn dataset-specific artifacts that create spurious correlations and weaken generalization. The experiment tested whether a model could predict the source corpus of an email, rather than predicting whether the email was phishing or legitimate. This is important because if a simple TF-IDF model can accurately identify the dataset source, it means the corpora contain strong dataset-specific artifacts. These artifacts may include source names, dates, header patterns, formatting, institutional terms, or topic-specific language. The dataset already shows that the corpora differ in structure and class composition, with some corpora being binary and others phishing-only. A balanced source-identification dataset was created using equal samples from each corpus. The task was defined as source dataset prediction, with the corpus name used as the target label rather than the phishing label. The dataset included six source corpora: CEAS_08, Enron, Ling, Nazario, Nigerian_Fraud, and SpamAssassin. From each corpus, 200 emails were sampled, producing a total sample size of 1200 emails. A 70/30 stratified train–test split was applied, resulting in a test set of 360 emails, and the random state was fixed at 42 for reproducibility. The model used the available raw text fields from each corpus and the text was capped to the first 500 characters per email for computational consistency. The parameters used by both the models are given below in Table 10.

Table 10. Model parameters: Dataset-identification experiment.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Task	Predict source corpus	Predict source corpus
Classifier	Logistic Regression	Linear SVC
Feature representation	TF-IDF	TF-IDF
N-gram range	Unigram	Unigram
Maximum TF-IDF features	1200	1200
Minimum document frequency	2	2
Sublinear TF scaling	True	True
Lowercasing	True	True
Accent stripping	Unicode	Unicode
Text length cap	First 500 characters	First 500 characters
Solver/implementation	lbfgs	liblinear-based Linear SVC
C	1.0	1.0
Max iterations	300	3000
Random state	42	42

Note: Logistic Regression used lbfgs here because this is a multi-class classification task with six corpus labels. The liblinear solver is mainly used for binary or one-vs.-rest settings and is less suitable for this direct multi-class setup.

3.10. Top-Feature Artifact Analysis

This step was conducted to further test H4 by examining whether the models relied on corpus-specific or artifact-like tokens when identifying dataset source and phishing class. The analysis focused on the most influential TF-IDF features used by the models. Specifically, the Linear SVC TF-IDF model was used for source-identification analysis, while the Logistic Regression TF-IDF model was used for phishing-classification analysis. Both models used TF-IDF feature representation with a maximum of 3000 features and incorporated raw text fields where available. The feature-inspection process focused on the highest positive coefficients to identify the terms most strongly associated with dataset source or phishing classification. These features were then grouped into artifact categories, including corpus markers, email header or MIME artifacts, URL/action tokens, boilerplate or list artifacts, and content tokens. By inspecting these high-weighted features, the study assessed whether model decisions were influenced by genuine phishing indicators or by dataset-specific artifacts such as corpus markers, header terms, date tokens, URL patterns, formatting cues, and other source-related textual signals.

3.11. Artifact-Removal Ablation Analysis

This analysis was conducted to further test H4 by examining whether removing or masking dataset-specific artifacts changes in-dataset and cross-dataset model performance. The purpose was to assess whether the models relied on genuine phishing indicators or on artifact-like features such as headers, URLs, corpus markers, formatting cues, and source-specific terms. To evaluate this, three text versions were compared. The first version was raw text, which included the subject, body, and available metadata fields. The second version was cleaned body-only text, where header-like fields, URLs, and MIME/HTML artifacts were reduced or removed. The third version was artifact-masked raw text, where source-specific and artifact-related terms were replaced with neutral placeholders. Changes in model performance across these three versions were used to determine whether artifact reliance influenced phishing detection robustness. The parameters used by the models are given below in Table 11.

Table 11. Model parameters: Artifact-removal ablation analysis.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Feature representation	TF-IDF	TF-IDF
N-gram range	Unigram	Unigram
Maximum TF-IDF features	2000	2000
Minimum document frequency	2	2
Text length cap	First 800 characters	First 800 characters
Logistic Regression Solver	liblinear	Not applicable
Linear SVC Implementation	Not applicable	liblinear-based Linear SVC
Random state	42	42

3.12. Artifact Transformation and Feature-Sensitivity Experiments

3.12.1. Adversarial Artifact Transformation

This experiment was conducted to further test H4 by examining whether phishing classifiers that rely on dataset-specific or surface-level textual artifacts can be affected by simple attacker-style transformations. The purpose was not only to test direct evasion, but also to assess whether the models' detection behavior changes when attacker-editable features are modified. These transformations included removing urgency or action words, replacing URLs or domain tokens, adding benign corporate footer text, removing header fields, normalizing case, and applying combined evasion changes, as shown in Table 12.

Table 12. Transformation applied: Adversarial artifact transformation.

Transformation	Purpose
Remove urgency/action words	Tests reliance on words such as “urgent,” “verify,” “click,” and “account.”
Replace URLs/domains	Tests reliance on domain or URL tokens such as .com, org, and link-like structures.
Add benign corporate footer	Tests on whether benign-looking boilerplate can shift model behavior.
Remove header fields	Tests reliance on metadata/header-style artifacts.
Normalize case	Tests reliance on formatting/capitalization.
Combined evasion	Application of multiple surface transformations together.

The adversarial artifact test was applied to the two main phishing classifiers used in the study (see Table 13 below). Detection rate means the proportion of phishing emails still classified as phishing after the transformation.

Table 13. Model parameters: Adversarial artifact transformation.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Feature representation	TF-IDF	TF-IDF
Classifier	Logistic Regression	Linear SVC
Text type	Phishing email text	Phishing email text
Attack target	Detection rate on phishing emails	Detection rate on phishing emails
Evaluation metric	Detection rate	Detection rate
Main comparison	Original phishing emails vs. transformed phishing emails	Original phishing emails vs. transformed phishing emails

3.12.2. Targeted Feature-Sensitivity Analysis

This experiment was conducted to test H4 by examining whether model performance declines when the model's own high-weight corpus-specific or phishing-associated features are removed or masked. The binary classification task was defined as phishing versus legitimate email, using a 70/30 stratified train–test split to preserve the class distribution. The test set contained 4800 emails, consisting of 2400 phishing emails and 2400 legitimate emails, with the random state fixed at 42 to ensure reproducibility. The parameters used

by both models are reported in Table 14, while the masking conditions compared in the experiment are presented in Table 15.

Table 14. Model parameters: Targeted feature-sensitivity analysis.

Parameter	Logistic Regression TF-IDF	Linear SVC TF-IDF
Feature representation	TF-IDF	TF-IDF
Classifier	Logistic Regression	Linear SVC
N-gram range	Unigram	Unigram
Maximum TF-IDF features	2000	2000
Minimum document frequency	2	2
Text length cap	First 800 characters	First 800 characters
Solver/implementation	liblinear	liblinear-based Linear SVC
C	1.0	1.0
Max iterations	300	3000
Random state	42	42

Table 15. Masking conditions: Targeted feature-sensitivity analysis.

Condition	Description
Original	No masking applied
Generic urgency/action masking	Masks common action terms such as urgent, verify, click, account, password, update, login, free, and money
Targeted feature-sensitivity masking	Masks corpus/source, URL/domain, date, action, and header-like artifact terms
Targeted artifact + URL/domain masking	Applies targeted feature-sensitivity masking plus URL/domain pattern replacement
Top-20 positive feature masking	Masks the top-20 phishing-associated features learned by each model
Top-50 positive feature masking	Masks the top-50 phishing-associated features learned by each model
Artifact + top-50 combined masking	Masks both targeted artifact terms and the top-50 phishing-associated model features

The top-50 positive phishing-associated features were extracted separately from the Logistic Regression TF-IDF and Linear SVC TF-IDF models. For the Logistic Regression TF-IDF model, the highest-ranked positive features were: your, you, gvc, ceas, monkey, challenge, user2, here, cnn, life, jose, our, user7, love, 2005, stock, user8, info, money, 300, account, 2004, watches, cc, top, site, watch, replica, company, bank, health, dear, transfer, hello, low, man, dollars, viagra, video, country, fast, price, best, sex, meds, we, huge, investment, click, and 2003. For the Linear SVC TF-IDF model, the highest-ranked positive features were: 2005, life, your, recipients, monkey, 2018, 2004, 300, 2003, you, stock, webmaster, user2, info, huge, jose, gvc, ceas, cnn, transfer, love, site, user7, user8, remove, jun, stop, marketing, hello, 2016, investment, proven, fast, medical, watch, health, em, es, within, government, cd, current, sir, body, cc, permanent, girls, here, usaa, and form.

The targeted top-20 and top-50 feature masking conditions were not intended to simulate a fully realistic external attacker. A real spammer would not necessarily know the classifier's highest-weighted features or systematically remove them across emails. Instead, these conditions were used as a white-box feature-sensitivity analysis to test whether model predictions depended heavily on a small set of learned textual cues. Therefore, the results should be interpreted as diagnostic evidence of feature reliance and artifact sensitivity, rather than as direct evidence of real-world adversarial evasion.

3.12.3. Artifact Masking Control Analysis

To clarify whether the improvement from artifact removal reflected artifact-specific effects or general noise reduction, an additional masking control was added. The artifact-removal condition was compared with a random-token-masking condition. In the random-masking condition, the same number of tokens removed or masked in the artifact condition was randomly selected from the model vocabulary, excluding stop words and label-defining

terms where possible. This ensured that the comparison controlled for information removed from the text.

The purpose of this analysis was to distinguish artifact-specific sensitivity from general text denoising. If artifact masking produced a larger cross-dataset performance change than random masking, this would provide stronger evidence that source-specific artifacts influenced model behavior. If random masking produced a similar effect, the improvement would be interpreted more cautiously as possible noise reduction rather than artifact-specific correction.

3.13. Statistical Validation

To determine whether the observed performance differences between the best single-corpus models and the pooled leave-one-corpus-out (LOCO) models were statistically reliable, paired statistical validation was added to the evaluation framework. Because each competing model was evaluated on the same held-out test corpus, paired bootstrap confidence intervals were used to estimate uncertainty around the difference in F1-score. For each comparison, the held-out test set was resampled with replacement 1000 times, and $\Delta F1$ was recalculated for each bootstrap sample. A difference was treated as statistically reliable when the 95% bootstrap confidence interval did not include zero. In addition, McNemar's test was applied to the paired binary predictions from the competing models on the same test instances. This test examined whether the two models made significantly different classification errors. McNemar's test was reported using a significance threshold of $\alpha = 0.05$. Together, the bootstrap confidence intervals and McNemar's test were used to distinguish reliable LOCO improvements from differences that may fall within the range of random variation.

3.14. Text Truncation Audit and Sensitivity Design

Several experiments used capped email text lengths to maintain computational consistency across corpora and experimental settings. Because truncation may remove later-occurring phishing cues, such as signatures, embedded links, disclaimers, footer content, or repeated URL patterns, a text truncation audit was added to assess the extent of information loss. For each email, the full character length was calculated before truncation. The proportion of emails exceeding the 500-, 800-, and 1000-character caps was then recorded. In addition, the mean number of characters removed was calculated among emails that exceeded each cap.

To examine whether text truncation affected the main findings, a sensitivity analysis was conducted using alternative text-length settings. The primary 1000-character setting was compared with a 2000-character setting and full-text input. The same leave-one-corpus-out (LOCO) evaluation design, classifiers, train-test structure, and evaluation metric were retained, while only the text-length setting was changed. Logistic Regression TF-IDF and Linear SVC TF-IDF were then re-evaluated using mean LOCO F1. This analysis was used to determine whether longer email representations materially changed the conclusion that pooled LOCO training improves cross-corpus robustness.

4. Results

4.1. Baseline Pooled Model Analysis

Table 16 shows that the pooled Logistic Regression model achieved the strongest F1-score (0.8091) among the three baseline models. This shows that pooled training can produce a strong phishing detection baseline. However, this result partially proves H1 because a pooled random split may still contain similar corpus patterns in both training and testing data.

Table 16. Pooled model analysis.

Model	Accuracy	Precision	Recall	F1	MCC	ROC-AUC	PR-AUC
Logistic Regression TF-IDF	0.7565	0.6831	0.9921	0.8091	0.5733	0.8688	0.8378
Linear SVC TF-IDF	0.7333	0.9904	0.4918	0.6572	0.5555	0.8710	0.8395

4.2. Single-Corpus Cross-Dataset Transfer Performance

This step tested whether a model trained on one corpus only can generalize to another unseen corpus. This is important because real phishing detection systems are often trained on one dataset but deployed in another environment. Each model was trained on one dataset and tested on a different dataset. Table 17 shows that single-corpus models produced unstable cross-dataset transfer performance. Although some source–target combinations performed strongly, such as Ling → CEAS_08 using Linear SVC with an F1-score of 0.848 and SpamAssassin → Ling using Logistic Regression with an F1-score of 0.804, other combinations showed substantial degradation. For example, CEAS_08 → Ling achieved only 0.394 F1 using Logistic Regression and 0.442 F1 using Linear SVC. Similarly, CEAS_08 → SpamAssassin produced weak F1-scores of 0.508 and 0.553 for Logistic Regression and Linear SVC, respectively. This variation indicates that models trained on a single corpus do not generalize consistently to unseen corpora. The precision–recall patterns further demonstrate this instability. Enron-trained models achieved very high recall when tested on SpamAssassin, with recall values of 0.992 for Logistic Regression and 0.994 for Linear SVC. However, these models also produced very low precision values of 0.385 and 0.402, together with high false-positive rates of 0.667 and 0.621. This indicates that the models detected most phishing emails but incorrectly flagged many legitimate emails as phishing. Conversely, CEAS_08-trained models tested on Ling showed the opposite problem, with low recall values of 0.286 and 0.347 and high false-negative rates of 0.714 and 0.653, meaning that many phishing emails were missed. These findings support H1 by establishing that single-corpus training leads to unstable cross-dataset performance. The wide variation in F1, MCC, FPR, and FNR suggests that models trained on one corpus often learn corpus-specific patterns rather than generalizable phishing indicators. Therefore, Table 17 provides the baseline justification for pooled multi-corpus training: if single-corpus models are unstable across unseen datasets, then combining multiple corpora during training is expected to improve robustness and produce more sustained performance across deployment environments.

Table 17. Single-corpus cross-dataset transfer.

Train Dataset	Test Dataset	Model	Precision	Recall	F1	MCC	FPR	FNR	ROC-AUC	PR-AUC
Enron	Ling	Logistic Regression TF-IDF	0.475	0.978	0.639	0.602	0.207	0.022	0.976	0.900
Enron	Ling	Linear SVC TF-IDF	0.403	0.959	0.568	0.520	0.271	0.041	0.954	0.838
Enron	SpamAssassin	Logistic Regression TF-IDF	0.385	0.992	0.554	0.350	0.667	0.008	0.942	0.880
Enron	SpamAssassin	Linear SVC TF-IDF	0.402	0.994	0.573	0.384	0.621	0.006	0.938	0.866
Ling	CEAS_08	Logistic Regression TF-IDF	0.877	0.590	0.706	0.498	0.104	0.410	0.853	0.858
Ling	CEAS_08	Linear SVC TF-IDF	0.839	0.857	0.848	0.651	0.207	0.143	0.875	0.872
Ling	Enron	Logistic Regression TF-IDF	0.884	0.548	0.676	0.532	0.063	0.452	0.844	0.847
Ling	Enron	Linear SVC TF-IDF	0.798	0.778	0.788	0.605	0.174	0.222	0.870	0.870
Ling	SpamAssassin	Logistic Regression TF-IDF	0.701	0.775	0.736	0.619	0.139	0.225	0.876	0.838
Ling	SpamAssassin	Linear SVC TF-IDF	0.531	0.886	0.664	0.509	0.328	0.114	0.905	0.856
SpamAssassin	CEAS_08	Logistic Regression TF-IDF	0.919	0.553	0.690	0.517	0.061	0.447	0.884	0.887
SpamAssassin	CEAS_08	Linear SVC TF-IDF	0.912	0.678	0.778	0.600	0.082	0.322	0.864	0.892
SpamAssassin	Enron	Logistic Regression TF-IDF	0.772	0.692	0.730	0.516	0.181	0.308	0.837	0.835
SpamAssassin	Enron	Linear SVC TF-IDF	0.692	0.812	0.747	0.494	0.320	0.188	0.832	0.830
SpamAssassin	Ling	Logistic Regression TF-IDF	0.762	0.852	0.804	0.766	0.051	0.148	0.963	0.903
SpamAssassin	Ling	Linear SVC TF-IDF	0.560	0.924	0.697	0.653	0.139	0.076	0.964	0.886
CEAS_08	Enron	Logistic Regression TF-IDF	0.843	0.660	0.741	0.571	0.109	0.340	0.862	0.861
CEAS_08	Enron	Linear SVC TF-IDF	0.829	0.665	0.738	0.560	0.121	0.335	0.852	0.836
CEAS_08	Ling	Logistic Regression TF-IDF	0.633	0.286	0.394	0.360	0.032	0.714	0.857	0.535
CEAS_08	Ling	Linear SVC TF-IDF	0.607	0.347	0.442	0.387	0.043	0.653	0.848	0.528
CEAS_08	SpamAssassin	Logistic Regression TF-IDF	0.678	0.406	0.508	0.388	0.081	0.594	0.863	0.694
CEAS_08	SpamAssassin	Linear SVC TF-IDF	0.754	0.436	0.553	0.456	0.060	0.564	0.876	0.734

Note: F1 = F1-score; MCC = Matthew's correlation coefficient; FPR = false-positive rate; FNR = false-negative rate; ROC-AUC = area under the receiver operating characteristic curve; PR-AUC = area under the precision–recall curve. The table reports cross-dataset transfer results, where each model was trained on one corpus and evaluated on a different corpus.

4.3. Leave-One-Corpus-Out (LOCO) Pooled Training

For each binary corpus, one corpus was held out as the test set. The model was trained on all other available corpora.

4.3.1. LOCO Along Logistic Regression TF-IDF

Table 18 provides direct support for H1 by showing that leave-one-corpus-out pooled training produced consistently strong cross-dataset performance for Logistic Regression TF-IDF. Across the four held-out corpora, F1-scores remained high, ranging from 0.8201 on Enron to 0.8994 on CEAS_08. This contrasts with the single-corpus transfer results in Table 17, where F1-scores varied widely and dropped to 0.394 for CEAS_08 → Ling and 0.508 for CEAS_08 → SpamAssassin. Therefore, pooled multi-corpus training produced more sustained F1 performance across unseen corpora. The model also achieved strong recall across all held-out corpora, ranging from 0.8730 on Enron to 0.9494 on SpamAssassin. This indicates that the pooled model detected most phishing emails even when the test corpus was unseen during training. MCC values were also consistently strong, ranging from 0.6463 to 0.8067, suggesting better balanced classification performance than many single-corpus transfer settings. The false-positive results were more varied, with Ling showing a low FPR of 0.0512, CEAS_08 0.1137, SpamAssassin 0.1269, and Enron a higher FPR of 0.2266. This suggests that pooled training improves overall robustness and F1 stability, although false-positive control may still require threshold calibration for some deployment corpora. Taken together, Table 18 supports H1 by demonstrating that Logistic Regression TF-IDF trained using leave-one-corpus-out pooled data generalizes more consistently than single-corpus models. The results indicate that multi-corpus training reduces dependence on individual corpus-specific patterns and improves cross-dataset phishing detection robustness.

Table 18. LOCO along Logistic Regression TF-IDF.

Held-Out Test Corpus	Precision	Recall	F1	MCC	FPR	FNR	ROC-AUC	PR-AUC
CEAS_08	0.9081	0.8908	0.8994	0.7754	0.1137	0.1092	0.9543	0.9582
Enron	0.7732	0.8730	0.8201	0.6463	0.2266	0.1270	0.9101	0.9075
Ling	0.7726	0.9127	0.8368	0.8067	0.0512	0.0873	0.9766	0.8895
SpamAssassin	0.7586	0.9494	0.8433	0.7774	0.1269	0.0506	0.9751	0.9476

4.3.2. LOCO Along Linear SVC TF-IDF

Table 19 presents the leave-one-corpus-out pooled-training results for the Linear SVC TF-IDF model. This table directly supports H1 because the model was trained on multiple corpora while one corpus was held out as a completely unseen test set. Overall, the results show that pooled multi-corpus training produced stable cross-dataset performance. The Linear SVC model achieved F1-scores between 0.7607 and 0.8910 across all held-out corpora. This is much more consistent than the single-corpus transfer results in Table 17, where some F1-scores dropped much lower, such as 0.442 for CEAS_08 → Ling and 0.553 for CEAS_08 → SpamAssassin. Therefore, Table 19 shows that pooled training helped the model generalize better to unseen corpora. The strongest result was on CEAS_08, where the model achieved an F1-score of 0.8910, precision of 0.8950, recall of 0.8871, and MCC of 0.7550. This shows a strong balance between detecting phishing emails and avoiding false positives. The model also performed very well on SpamAssassin, with the highest recall of 0.9581 and the lowest false-negative rate of 0.0419, meaning it missed very few phishing emails in that held-out corpus. The weakest result was on Ling, with an F1-score of 0.7607. However, this is still a strong result compared with several single-corpus transfer outcomes in Table 17. The model also kept the false-positive rate relatively low on Ling

at 0.0875, which supports the idea that pooled training can reduce unstable false-positive behavior. Enron remained the most challenging corpus for false positives, with an FPR of 0.2881. This means the model incorrectly flagged more legitimate Enron emails as phishing. However, its F1-score was still 0.8024, showing that pooled training maintained useful overall performance despite this limitation. In brief, Table 19 supports H1 because Linear SVC TF-IDF trained with pooled multi-corpus data achieved consistently strong F1-scores, high recall, and solid MCC values across unseen corpora. The results suggest that pooled training improves cross-dataset robustness compared with single-corpus training, although false-positive control may still require threshold tuning for some datasets such as Enron.

Table 19. LOCO along Linear SVC TF-IDF.

Held-Out Test Corpus	Precision	Recall	F1	MCC	FPR	FNR	ROC-AUC	PR-AUC
CEAS_08	0.8950	0.8871	0.8910	0.7550	0.1313	0.1129	0.9400	0.9511
Enron	0.7318	0.8881	0.8024	0.6048	0.2881	0.1119	0.8918	0.8809
Ling	0.6613	0.8952	0.7607	0.7189	0.0875	0.1048	0.9587	0.8435
SpamAssassin	0.7571	0.9581	0.8458	0.7818	0.1291	0.0419	0.9776	0.9507

4.4. Direct Comparison: Pooled-Minus-Test vs. Best Single-Corpus Model

4.4.1. Logistic Regression TF-IDF Comparison

Table 20 supports H1 by showing that pooled-minus-test Logistic Regression TF-IDF outperformed the best single-corpus baseline on all held-out corpora in terms of F1 and MCC. F1 improvements ranged from +0.0327 on Ling to +0.1938 on CEAS_08, while MCC also increased across all corpora. These results indicate that pooled multi-corpus training improves cross-dataset robustness and produces more sustained classification performance than single-corpus training. However, false-positive-rate changes were mixed, suggesting that the false-positive component of H1 is threshold- and corpus-dependent.

Table 20. Pooled-minus-test Logistic Regression TF-IDF.

Test Corpus	Best Single Corpus	Best Single F1	Pooled-Minus-Test F1	Δ F1	Best Single MCC	Pooled MCC	Δ MCC	FPR Change
CEAS_08	Ling	0.7056	0.8994	+0.1938	0.4984	0.7754	+0.2770	+0.0095
Enron	CEAS_08	0.7406	0.8201	+0.0795	0.5709	0.6463	+0.0754	+0.1180
Ling	SpamAssassin	0.8041	0.8368	+0.0327	0.7660	0.8067	+0.0407	+0.0004
SpamAssassin	Ling	0.7362	0.8433	+0.1072	0.6190	0.7774	+0.1584	−0.0117

4.4.2. Linear SVC TF-IDF Comparison

Table 21 provides strong support for H1. The pooled-minus-test Linear SVC TF-IDF model improved F1 across all held-out corpora, with gains ranging from +0.0143 on Enron to +0.1816 on SpamAssassin. MCC also improved in three of the four corpora, showing better balanced classification performance in most unseen corpus settings. In addition, false-positive rates decreased for CEAS_08, Ling, and SpamAssassin, suggesting that pooled multi-corpus training can reduce false alarms for Linear SVC. However, the Enron result shows that false-positive control remains corpus-dependent. Overall, Table 21 supports H1 by demonstrating that pooled training improves cross-dataset robustness compared with single-corpus training.

Table 21. Pooled-minus-test Linear SVC TF-IDF comparison.

Test Corpus	Best Single Corpus	Best Single F1	Pooled-Minus-Test F1	$\Delta F1$	Best Single MCC	Pooled MCC	ΔMCC	FPR Change
CEAS_08	Ling	0.8478	0.8910	+0.0433	0.6514	0.7550	+0.1036	−0.0760
Enron	Ling	0.7881	0.8024	+0.0143	0.6052	0.6048	−0.0005	+0.1141
Ling	SpamAssassin	0.6969	0.7607	+0.0638	0.6528	0.7189	+0.0661	−0.0512
SpamAssassin	Ling	0.6642	0.8458	+0.1816	0.5090	0.7818	+0.2728	−0.1992

4.5. Prevalence-Shift Simulation

4.5.1. Simulation for Logistic Regression TF-IDF

Table 22 presents the prevalence-shift simulation results for the high-prevalence-trained phishing detection model under fixed-threshold and operational alert-budget settings. Under the fixed-threshold condition, recall remained consistently high across all test prevalence levels, ranging from 0.951 to 0.960 for test environments with 1% to 10% phishing prevalence. This indicates that low deployment prevalence alone did not cause recall collapse. However, this high recall was achieved at the cost of substantial false-positive burden. At 1% phishing prevalence, the model generated approximately 2722 alerts per 10,000 emails, with a false-positive rate of 0.256, which would be operationally difficult to manage in a real organizational email environment. When alert-budget thresholds were applied, recall decreased substantially. At 1% phishing prevalence, recall declined from 0.960 under the fixed threshold to 0.230 under the top-1% alert-budget condition. Although the top-1% alert budget reduced the false-positive rate from 0.256 to 0.005 and reduced alerts from 2722 to approximately 101 per 10,000 emails, this came at the cost of missing most phishing emails. A similar trade-off was observed under the top-5% and top-10% alert-budget conditions, where recall improved relative to the top-1% condition but remained lower than the fixed-threshold setting. The results therefore show that recall collapse emerges when low-prevalence deployment is combined with stricter operational thresholding or alert-budget constraints, which is H2.

Table 22. Prevalence-shift simulation for Logistic Regression TF-IDF.

Test Phishing Prevalence	Threshold Rule	Precision	Recall	F1	FPR	Alerts per 10,000
1%	Fixed threshold	0.100	0.960	0.170	0.256	2722
1%	Top-1% alert budget	0.476	0.230	0.289	0.005	101
1%	Top-5% alert budget	0.290	0.668	0.378	0.036	501
1%	Top-10% alert budget	0.180	0.825	0.282	0.084	1001
5%	Fixed threshold	0.303	0.955	0.429	0.256	3243
5%	Top-1% alert budget	0.860	0.111	0.193	0.002	100
5%	Top-5% alert budget	0.703	0.434	0.507	0.016	500
5%	Top-10% alert budget	0.586	0.673	0.589	0.044	1001
10%	Fixed threshold	0.399	0.951	0.545	0.258	3570
10%	Top-1% alert budget	0.943	0.073	0.135	0.001	100
10%	Top-5% alert budget	0.859	0.327	0.466	0.008	500
10%	Top-10% alert budget	0.759	0.564	0.634	0.027	1001
20%	Fixed threshold	0.537	0.953	0.680	0.257	4133
20%	Top-1% alert budget	0.968	0.045	0.085	0.000	101
20%	Top-5% alert budget	0.946	0.217	0.351	0.003	501
20%	Top-10% alert budget	0.905	0.413	0.564	0.012	1000
50%	Fixed threshold	0.797	0.954	0.867	0.253	6035
50%	Top-1% alert budget	0.988	0.020	0.040	0.000	103
50%	Top-5% alert budget	0.992	0.099	0.181	0.001	501
50%	Top-10% alert budget	0.988	0.198	0.330	0.003	1001

Figure 2 shows that recall for Logistic Regression TF-IDF remains consistently high under the fixed-threshold setting, staying around 0.95 across all prevalence levels. In

contrast, recall drops sharply when alert-budget constraints are imposed. The stricter the budget, the larger the decline: the top-1% alert budget produces the lowest recall, while the top-10% budget preserves more recall but still declines steadily as prevalence increases. This supports H2 by showing that recall collapse is driven not by prevalence shift alone, but by the interaction between prevalence shift and limited alert capacity.

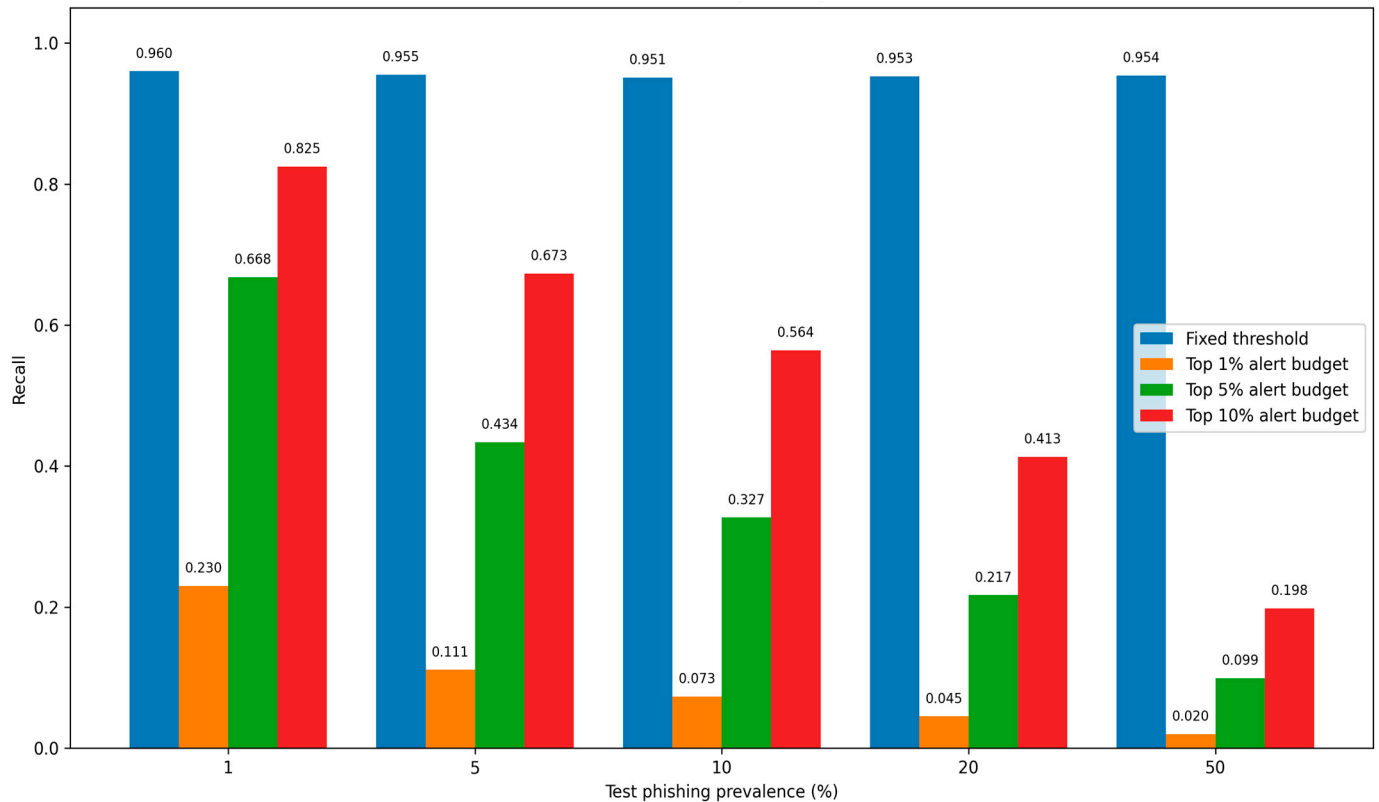


Figure 2. Prevalence-shift recall Logistic Regression TF-IDF.

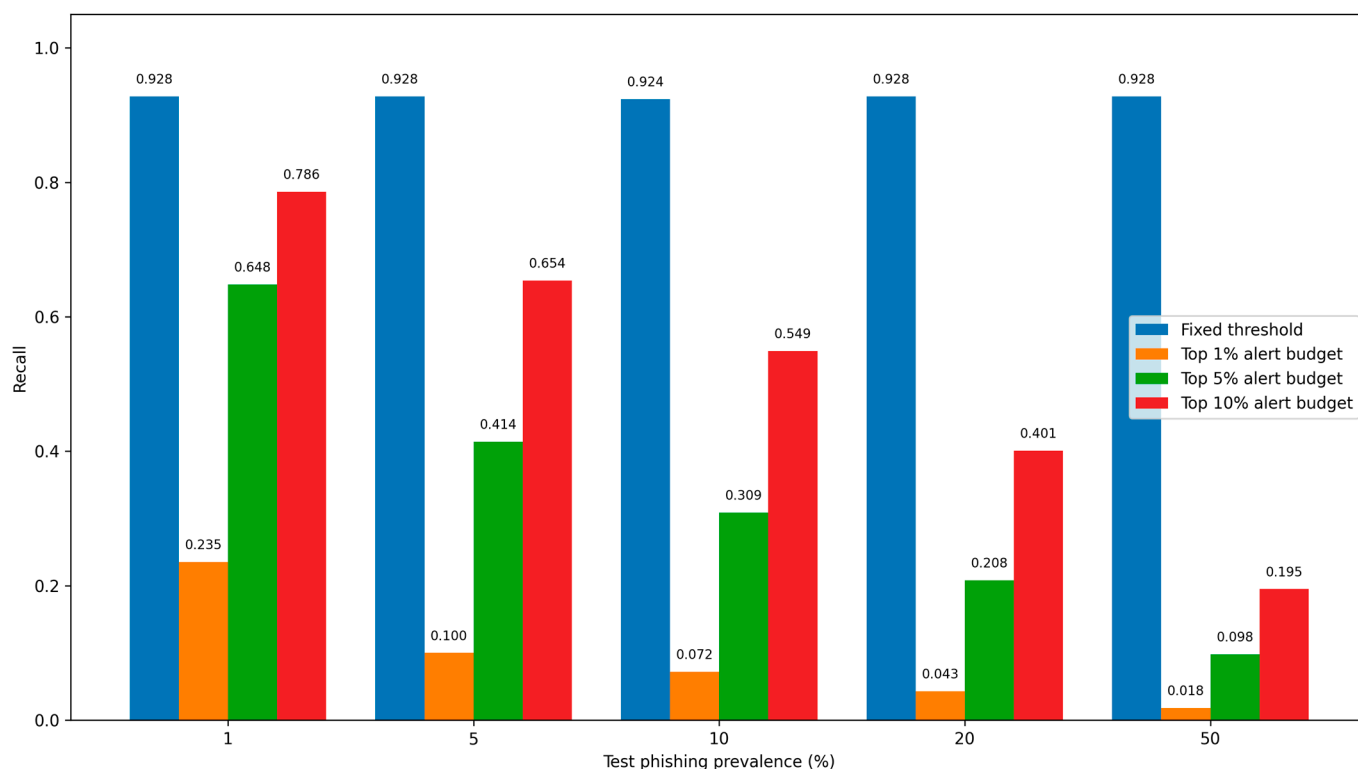
4.5.2. Simulation for Linear SVC TF-IDF

Table 23 presents the H2 prevalence-shift results for the Linear SVC TF-IDF model trained under a high-phishing-prevalence setting and evaluated under different deployment prevalence and threshold conditions. Under the fixed-threshold setting, recall remained consistently high across all test prevalence levels, ranging from 0.922 to 0.928. This indicates that low phishing prevalence alone did not cause recall collapse. However, this high recall came with a substantial false-positive burden. For example, at 1% phishing prevalence, the model generated approximately 2346 alerts per 10,000 emails, with an FPR of 0.219, which would be difficult to manage in a real organizational email system. When operational alert-budget thresholds were applied, recall dropped sharply. At 1% phishing prevalence, recall decreased from 0.928 under the fixed threshold to 0.235 under the top-1% alert-budget condition. Although this reduced the FPR from 0.219 to 0.005 and limited alerts to approximately 101 per 10,000 emails, it also meant that most phishing emails were missed. The same pattern appears across other prevalence levels: stricter alert budgets produce higher precision and lower FPR but substantially reduce recall. Therefore, Table 23 supports H2 by showing that recall collapse is not caused by low-prevalence deployment alone. Instead, recall collapse occurs when a high-prevalence-trained model is deployed under low-prevalence, alert-limited operational conditions. In practical terms, controlling false-positive workload through strict alert budgets can severely reduce phishing detection coverage.

Table 23. Prevalence-shift simulation for Linear SVC TF-IDF.

Test Phishing Prevalence	Threshold Rule	Precision	Recall	F1	FPR	Alerts per 10,000
1%	Fixed threshold	0.111	0.928	0.186	0.219	2346
1%	Top-1% alert budget	0.491	0.235	0.298	0.005	101
1%	Top-5% alert budget	0.295	0.648	0.380	0.036	501
1%	Top-10% alert budget	0.176	0.788	0.275	0.084	1001
5%	Fixed threshold	0.330	0.928	0.456	0.220	2892
5%	Top-1% alert budget	0.805	0.100	0.174	0.002	100
5%	Top-5% alert budget	0.702	0.415	0.493	0.016	500
5%	Top-10% alert budget	0.588	0.653	0.583	0.044	1001
10%	Fixed threshold	0.431	0.922	0.569	0.221	3216
10%	Top-1% alert budget	0.930	0.072	0.132	0.001	100
10%	Top-5% alert budget	0.834	0.312	0.447	0.009	500
10%	Top-10% alert budget	0.751	0.549	0.621	0.028	1001
20%	Fixed threshold	0.568	0.928	0.696	0.222	3809
20%	Top-1% alert budget	0.980	0.045	0.086	0.000	101
20%	Top-5% alert budget	0.918	0.210	0.340	0.005	501
20%	Top-10% alert budget	0.887	0.404	0.551	0.014	1000
50%	Fixed threshold	0.818	0.926	0.867	0.216	5710
50%	Top-1% alert budget	0.995	0.021	0.040	0.000	103
50%	Top-5% alert budget	0.990	0.099	0.180	0.001	501
50%	Top-10% alert budget	0.979	0.196	0.327	0.004	1001

Figure 3 shows that recall for Linear SVC TF-IDF remains almost unchanged under the fixed-threshold setting, staying around 0.92–0.93 across all phishing prevalence levels. However, recall drops sharply when alert-budget constraints are imposed, especially under the top-1% alert budget. The top-5% and top-10% budgets preserve more recall, but both still decline steadily as phishing prevalence increases. This supports H2 by showing that recall collapse is caused by operational alert-budget limits rather than by prevalence shift alone.

**Figure 3.** Prevalence-shift recall Linear SVC TF-IDF.

4.6. Training Prevalence Manipulation

4.6.1. Manipulation for Logistic Regression TF-IDF

Table 24 shows that with 5-phishing-prevalence training, the model achieved perfect precision of 1.0000, but recall was only 0.2920. This means the model rarely produced false positives, but it missed most phishing emails in the high-threat deployment environment. The false-negative rate was very high at 0.7080, meaning 70.8% of phishing emails were missed. As training phishing prevalence increased, recall improved sharply. This shows that low-prevalence training made the model overly conservative, which supports H3. The model became much better at detecting phishing emails only after the training data included more phishing examples. Calibration also improved as training prevalence increased from 5% to 50%. At 5% training prevalence, calibration error was 0.3046, but at 50% training prevalence it decreased to 0.0936. This supports the calibration part of H3.

Table 24. Training prevalence manipulation: Logistic Regression TF-IDF.

Training Phishing Prevalence	Deployment Phishing Prevalence	Precision	Recall	F1	MCC	FPR	FNR	ROC-AUC	PR-AUC	Calibration Error
5%	50%	1.0000	0.2920	0.4520	0.4135	0.0000	0.7080	0.9837	0.9831	0.3046
10%	50%	0.9946	0.5560	0.7133	0.6162	0.0030	0.4440	0.9881	0.9877	0.2107
30%	50%	0.9851	0.8910	0.9357	0.8815	0.0135	0.1090	0.9930	0.9925	0.1177
50%	50%	0.9635	0.9645	0.9640	0.9280	0.0365	0.0355	0.9932	0.9927	0.0936
70%	50%	0.9174	0.9940	0.9542	0.9077	0.0895	0.0060	0.9931	0.9923	0.1052

Figure 4 shows that at 5% training phishing prevalence, precision is almost perfect, but recall is exceptionally low. This means the model rarely misclassifies legitimate emails as phishing, but it misses many real phishing emails. As training phishing prevalence increases from 5% to 50%, recall rises sharply and F1 also improves. This shows that the model becomes better at detecting phishing emails when it is exposed to more examples of phishing during training. The key pattern is that precision does not collapse. Instead, recall is the main weakness under low-prevalence training. Therefore, the bar chart supports H3.

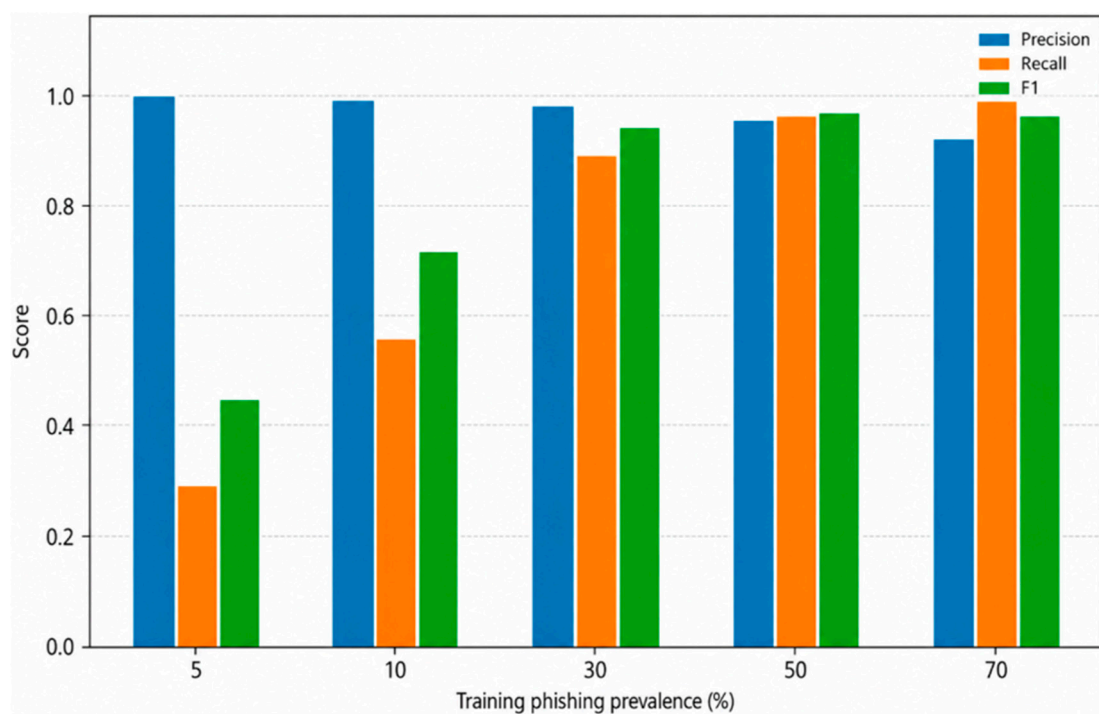


Figure 4. Training prevalence shift: Logistic Regression TF-IDF.

4.6.2. Manipulation for Linear SVC TF-IDF

Table 25 shows that for Linear SVC TF-IDF, the same general pattern was found, although the recall collapse is less severe than with Logistic Regression. With 5-phishing-prevalence training, recall was 0.6315, meaning the model missed 36.85% of phishing emails in the high-threat deployment environment. Precision was remarkably high at 0.9961, meaning the model was highly conservative and produced very few false positives. As training prevalence increased, recall improved. This supports H3 because the model trained under low phishing prevalence was less able to detect phishing emails when the deployment environment contained many phishing messages.

Table 25. Training prevalence manipulation: Linear SVC TF-IDF.

Training Phishing Prevalence	Deployment Phishing Prevalence	Precision	Recall	F1	MCC	FPR	FNR	ROC-AUC	PR-AUC
5%	50%	0.9961	0.6315	0.7729	0.6759	0.0025	0.3685	0.9856	0.9864
10%	50%	0.9925	0.7930	0.8816	0.8034	0.0060	0.2070	0.9903	0.9908
30%	50%	0.9807	0.9385	0.9591	0.9209	0.0185	0.0615	0.9944	0.9935
50%	50%	0.9653	0.9725	0.9689	0.9375	0.0350	0.0275	0.9948	0.9938
70%	50%	0.9489	0.9925	0.9702	0.9400	0.0535	0.0075	0.9948	0.9928

Figure 5 shows that at 5% training phishing prevalence, precision is remarkably high (0.9961), but recall is much lower (0.6315). This means the model makes very few false-positive errors, but it misses a considerable number of phishing emails. As the training phishing prevalence increases, recall improves steadily from 0.6315 at 5% to 0.9925 at 70%, while F1 also rises from 0.7729 to 0.9702. This pattern shows that the precision collapse does not occur. Instead, the main failure is recalling degradation under low-prevalence training. Therefore, the figure supports that models trained with too few phishing examples become overly cautious and miss phishing emails when deployed in high-prevalence threat environments, which supports H3.

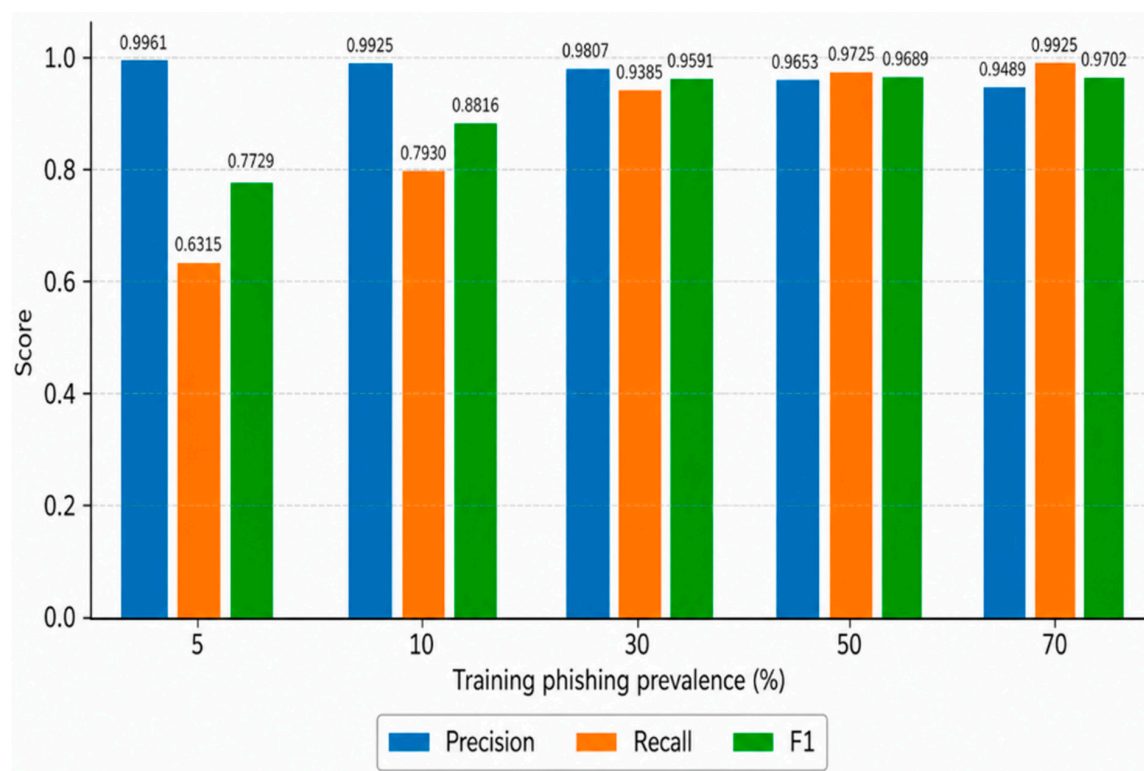


Figure 5. Training prevalence shift: Linear SVC TF-IDF.

4.7. Dataset Artifacts and Spurious Correlations

4.7.1. Dataset-Identification Performance

Tables 26 and 27 strongly support H4. The results showed that dataset identity was highly learnable from the email text. Logistic Regression TF-IDF achieved an accuracy of 0.9722 and macro-F1 of 0.9723, while Linear SVC TF-IDF achieved an accuracy of 0.9806 and macro-F1 of 0.9805. Per-corpus F1-scores were also high across all datasets, ranging from 0.9440 to 0.9917 for Logistic Regression and from 0.9587 to 1.0000 for Linear SVC. These findings support H4 by showing that the corpora contain strong source-specific signals. Therefore, phishing classifiers trained on these datasets may learn corpus identity or formatting patterns in addition to genuine phishing indicators. This provides evidence that dataset-specific artifacts can create spurious correlations, helping explain why models perform strongly in benchmark settings but show weaker cross-dataset generalization.

Table 26. Dataset-identification performance.

Task	Model	Accuracy	Macro F1	Interpretation
Predict source dataset	Logistic Regression TF-IDF	0.9722	0.9723	Very high score; dataset artifacts are strongly learnable
Predict source dataset	Linear SVC TF-IDF	0.9806	0.9805	Very high score; dataset artifacts are strongly learnable

Table 27. Dataset-identification performance per corpus F1.

Corpus	Logistic Regression TF-IDF F1	Linear SVC TF-IDF F1
CEAS_08	0.9744	0.9917
Enron	0.9565	0.9655
Ling	0.9440	0.9587
Nazario	0.9836	0.9836
Nigerian_Fraud	0.9833	0.9833
SpamAssassin	0.9917	1.0000

4.7.2. Top Source Dataset Feature Analysis

Table 28 presents the highest-weighted features used by the dataset-identification classifier to distinguish source corpora. The results show that each corpus contains strongly identifiable textual markers. For example, CEAS_08 is associated with features such as 2008, ceas, gvc, and challenge, while Enron is associated with vince, enron, 2001, and meter. Ling is characterized by topic-specific tokens such as language, linguistics, and english, whereas Nazario and SpamAssassin contain corpus/source markers such as monkey and spamassassin. These features do not represent general phishing behavior; instead, they reflect corpus origin, time period, formatting, topic, or source-specific collection patterns. Therefore, Table 28 supports H4 by showing that dataset-specific artifacts are learnable and may create spurious correlations that weaken cross-dataset phishing detection generalization.

Table 28. Top source-identifying features by corpus.

Corpus	Rank	Feature	Coefficient	Likely Artifact Type
CEAS_08	1	2008	4.6125	Content/source token
CEAS_08	2	aug	2.4022	Content/source token
CEAS_08	3	ceas	1.5996	Corpus-specific marker
CEAS_08	4	gvc	1.5996	Content/source token
CEAS_08	5	challenge	1.5615	Content/source token
CEAS_08	6	5	1.5499	Content/source token
CEAS_08	7	6	1.3784	Content/source token

Table 28. Cont.

Corpus	Rank	Feature	Coefficient	Likely Artifact Type
CEAS_08	8	tue	1.3347	Content/source token
Enron	1	vince	1.5196	Content/source token
Enron	2	enron	1.2478	Corpus-specific marker
Enron	3	2001	1.0805	Content/source token
Enron	4	drugs	1.0763	Content/source token
Enron	5	paliourg	1.0676	Content/source token
Enron	6	meter	1.0139	Content/source token
Enron	7	2000	0.9347	Content/source token
Enron	8	rice	0.9103	Content/source token
Ling	1	language	1.7693	Content/source token
Ling	2	linguistics	1.5620	Corpus-specific marker
Ling	3	english	1.3040	Content/source token
Ling	4	1998	1.1342	Content/source token
Ling	5	the	1.0787	Content/source token
Ling	6	linguist	1.0590	Corpus-specific marker
Ling	7	bulk	1.0234	Content/source token
Ling	8	and	1.0146	Content/source token
Nazario	1	monkey	2.0964	Corpus-specific marker
Nazario	2	jose	2.0214	Content/source token
Nazario	3	2016	1.4360	Content/source token
Nazario	4	recipients	1.2664	Content/source token
Nazario	5	2017	1.2519	Content/source token
Nazario	6	account	1.2205	Content/source token
Nazario	7	org	1.2124	URL/domain artifact
Nazario	8	2018	1.1186	Content/source token
Nigerian_Fraud	1	2006	1.4458	Content/source token
Nigerian_Fraud	2	my	1.3828	Content/source token
Nigerian_Fraud	3	mr	1.3365	Content/source token
Nigerian_Fraud	4	2007	1.2346	Content/source token
Nigerian_Fraud	5	am	1.2112	Content/source token
Nigerian_Fraud	6	2005	1.0702	Content/source token
Nigerian_Fraud	7	yahoo	1.0197	Content/source token
Nigerian_Fraud	8	bank	0.9783	Content/source token
SpamAssassin	1	2002	6.2694	Content/source token
SpamAssassin	2	net	1.6714	Content/source token
SpamAssassin	3	taint	1.6565	Content/source token
SpamAssassin	4	com	1.6441	URL/domain artifact
SpamAssassin	5	sep	1.6073	Content/source token
SpamAssassin	6	jul	1.5723	Content/source token
SpamAssassin	7	spamassassin	1.5299	Corpus-specific marker
SpamAssassin	8	yyyy	1.2598	Content/source token

4.7.3. Top-Phishing-Classifer Feature Analysis

The phishing-classifier feature analysis provides further evidence of dataset-specific artifact reliance. Table 29 shows that several phishing-associated features, such as ceas, gvc, and challenge, appear to reflect CEAS_08 source or benchmark-specific patterns rather than general phishing behavior. Other phishing-associated terms, such as click, free, money, and site, are attacker-editable action or promotional terms that may not remain stable under adversarial rewriting. Similarly, several legitimate-associated features, including enron, language, university, edu, and re, appear to capture corpus identity, topic, domain, or reply-thread formatting rather than legitimacy in general. These results support H4 by showing that phishing classifiers can learn spurious correlations linked to corpus origin and formatting, which may inflate benchmark performance while weakening cross-dataset generalizability.

Table 29. Phishing-classifier high-weight-feature examples.

Rank	Feature	Associated Class	Coefficient	Likely Artifact Type	Why Problematic
1	your	Phishing	4.359	Content/action wording	Common persuasive wording; may reflect phishing style but is attacker-editable.
2	you	Phishing	4.070	Content/action wording	Broad second-person wording; not a stable phishing-specific marker.
3	free	Phishing	2.601	Promotional content token	May indicate scam-like language, but attackers can easily remove or replace it.
4	money	Phishing	1.979	Financial content token	Relevant to phishing but also appears in legitimate finance emails.
5	here	Phishing	1.841	Action/boilerplate wording	May reflect clickable call-to-action phrasing rather than phishing intent.
6	click	Phishing	1.710	URL/action token	Attacker-editable wording; models relying on this may fail if wording changes.
7	user2	Phishing	1.700	Corpus/source token	Likely anonymization- or dataset-specific formatting, not genuine phishing behavior.
8	our	Phishing	1.679	Content/boilerplate token	Broad wording can appear in both legitimate and phishing emails.
9	gvc	Phishing	1.660	Corpus/source token	May reflect CEAS_08-specific address or collection format.
10	ceas	Phishing	1.660	Corpus-specific marker	May identify the CEAS_08 corpus rather than phishing behavior.
11	challenge	Phishing	1.604	Corpus/source token	May reflect benchmark source or dataset context rather than attack intent.
12	com	Phishing	1.568	URL/domain artifact	Broad web/domain token; may reflect URL-heavy formatting rather than phishing.
13	best	Phishing	1.555	Promotional content token	May reflect marketing/scam style but is easily rewritten.
14	business	Phishing	1.506	Content/source token	Can occur in legitimate business emails as well as phishing.
15	we	Phishing	1.498	Content/boilerplate token	Generic pronoun; weak as a stable phishing indicator.
16	cnn	Phishing	1.458	Content/source token	May reflect dataset topic/source rather than phishing behavior.
17	site	Phishing	1.449	URL/action token	Related to web links but not necessarily phishing-specific.
18	company	Phishing	1.442	Content/source token	Common in legitimate and phishing emails; context-dependent.
19	remove	Phishing	1.424	Mailing/boilerplate token	May reflect unsubscribe or list-style text rather than phishing intent.
20	watch	Phishing	1.405	Content/action token	May reflect promotional or dataset-specific wording.
1	wrote	Legitimate	−2.938	Reply/thread artifact	May identify conversational email style rather than legitimacy generally.
2	org	Legitimate	−2.595	URL/source token	May reflect mailing-list or institutional domains rather than legitimate status.
3	enron	Legitimate	−2.554	Corpus-specific marker	May identify Enron as benign rather than legitimate emails generally.
4	user6	Legitimate	−2.481	Corpus/source token	Likely anonymization- or dataset-specific formatting.
5	language	Legitimate	−2.412	Corpus/topic marker	May reflect Ling corpus topic rather than genuine legitimacy.
6	thanks	Legitimate	−2.225	Conversational content token	Common polite wording; not a reliable legitimacy marker.
7	the	Legitimate	−2.224	Generic content token	Too general to represent meaningful legitimacy.
8	re	Legitimate	−2.190	Email reply artifact	May indicate reply-thread format rather than legitimate status.

Table 29. Cont.

Rank	Feature	Associated Class	Coefficient	Likely Artifact Type	Why Problematic
9	university	Legitimate	−2.027	Corpus/topic marker	May reflect academic/Ling corpus content rather than legitimacy generally.
10	2000	Legitimate	−1.969	Date/source token	May reflect Enron/Ling collection period rather than legitimate email behavior.
11	edu	Legitimate	−1.946	Domain/source token	May reflect educational or mailing-list sources rather than legitimacy.
12	on	Legitimate	−1.888	Generic content token	Too broad to be a stable legitimacy marker.

4.7.4. Artifact-Removal Ablation

The artifact-removal ablation strongly supports H4. In the raw-text condition, both models achieved high in-dataset F1s but much lower cross-dataset F1s. Table 30 shows that Logistic Regression dropped from a 0.9552 in-dataset F1 to a 0.5720 cross-dataset F1, while Linear SVC dropped from 0.9603 to 0.5934. This suggests that raw-text models learned dataset-specific patterns that did not transfer well to unseen corpora. After cleaning the body text and reducing artifacts, cross-dataset F1 improved. Logistic Regression increased from 0.5720 to 0.6485 (+0.0765), and Linear SVC increased from 0.5934 to 0.6779 (+0.0844). Although in-dataset performance slightly decreased, cross-dataset generalization improved, indicating that metadata, formatting, and corpus-specific tokens were inflating benchmark performance. Therefore, the results support H4 by showing that artifact reduction improves robustness and reduces reliance on spurious dataset-specific correlations.

Table 30. Artifact-removal ablation.

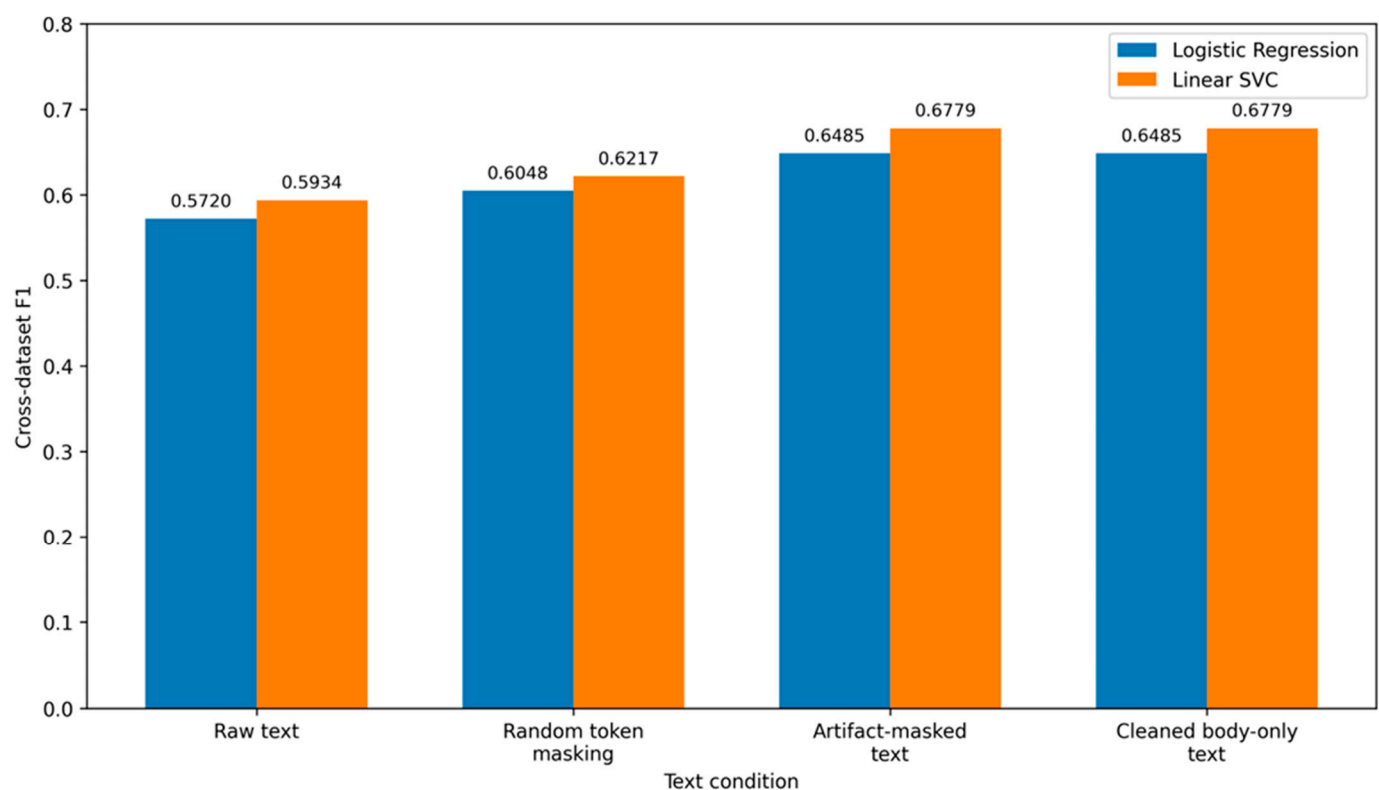
Text Version	Model	Evaluation	F1	MCC	FPR	FNR
Raw text	Logistic Regression TF-IDF	In-dataset split	0.9552	0.9127	0.0458	0.0413
Raw text	Logistic Regression TF-IDF	Cross-dataset transfer	0.5720	0.3721	0.3715	0.3150
Cleaned body only	Logistic Regression TF-IDF	In-dataset split	0.9450	0.8927	0.0569	0.0501
Cleaned body only	Logistic Regression TF-IDF	Cross-dataset transfer	0.6485	0.4295	0.3129	0.3039
Artifact-masked raw text	Logistic Regression TF-IDF	In-dataset split	0.9503	0.9041	0.0403	0.0560
Artifact-masked raw text	Logistic Regression TF-IDF	Cross-dataset transfer	0.5831	0.3838	0.3163	0.3500
Raw text	Linear SVC TF-IDF	In-dataset split	0.9603	0.9227	0.0403	0.0369
Raw text	Linear SVC TF-IDF	Cross-dataset transfer	0.5934	0.3986	0.3614	0.2974
Cleaned body only	Linear SVC TF-IDF	In-dataset split	0.9471	0.8970	0.0542	0.0487
Cleaned body only	Linear SVC TF-IDF	Cross-dataset transfer	0.6779	0.4500	0.3140	0.2750
Artifact-masked raw text	Linear SVC TF-IDF	In-dataset split	0.9558	0.9141	0.0431	0.0428
Artifact-masked raw text	Linear SVC TF-IDF	Cross-dataset transfer	0.6134	0.4096	0.3192	0.3196

Table 31 compares whether the improvement after artifact removal is due to artifact-specific effects or general noise reduction. Raw text produces the weakest cross-dataset F1 for both models, confirming poor baseline transfer performance. Random token masking improves F1 slightly, suggesting that removing some unstable or noisy text can help generalization. However, artifact-masked text produces a larger improvement, increasing F1 from 0.5720 to 0.6485 for Logistic Regression and from 0.5934 to 0.6779 for Linear SVC. This indicates that the gain is not only due to generic noise reduction but is strongly associated with reducing source-specific and artifact-like features. The cleaned body-only condition produces similar improvement, further supporting the interpretation that headers, source markers, URL/domain noise, and formatting artifacts contribute to weak cross-dataset generalization.

Table 31. Artifact masking control analysis.

Text Condition	LR Cross-Dataset F1	Linear SVC Cross-Dataset F1	Interpretation
Raw text	0.5720	0.5934	Baseline cross-dataset performance
Random token masking	0.6048	0.6217	Tests whether improvement is due to generic noise reduction
Artifact-masked text	0.6485	0.6779	Larger improvement suggests artifact-specific effect
Cleaned body-only text	0.6485	0.6779	Reduces headers, source markers, URL/domain noise, and formatting artifacts

Figure 6 shows that artifact-removal operations improve cross-dataset F1 for both models. Raw text produced the lowest F1, with 0.5720 for Logistic Regression and 0.5934 for Linear SVC. Random token masking produced a small improvement, suggesting that some gain may come from reducing noisy text. However, artifact-masked text and cleaned body-only text produced the highest F1-scores, reaching 0.6485 for Logistic Regression and 0.6779 for Linear SVC. This indicates that reducing corpus-specific artifacts, headers, URL/domain noise, and formatting cues improves cross-dataset generalization more than random masking alone.

**Figure 6.** Cross-dataset F1 under artifact-removal conditions.

4.8. Artifact Transformation and Feature Sensitivity

4.8.1. Adversarial Artifact Transformation

Table 32 shows that the simple adversarial transformations did not produce a major detection collapse. Both models maintained high detection rates after all transformations. For Logistic Regression TF-IDF, the original detection rate was 0.974 and transformed detection rates ranged from 0.972 to 0.990. For Linear SVC TF-IDF, the original detection rate was 0.978 and transformed detection rates ranged from 0.976 to 0.988. This means that simple surface-level edits, such as removing urgency words or replacing URLs, were not sufficient to evade these models, although the adversarial test shows that model behavior changes when artifact-like elements are modified. The clearest example is header removal,

which produced a small detection drop for both models, for Logistic Regression TF-IDF 0.974 to 0.972 and for Linear SVC TF-IDF 0.978 to 0.976. The drop is small, but it points in the expected direction that removing header-like artifacts slightly reduced detection. This is consistent with the earlier H4 results showing that metadata, formatting, and corpus-specific artifacts influence model behavior.

Table 32. Adversarial artifact transformation.

Model	Attack Type	Detection Rate	Detection Drop
Logistic Regression TF-IDF	Original phishing emails	0.974	0.000
Logistic Regression TF-IDF	Remove urgency/action words	0.974	0.000
Logistic Regression TF-IDF	Replace URLs/domains	0.976	0.002
Logistic Regression TF-IDF	Add benign corporate footer	0.986	0.012
Logistic Regression TF-IDF	Remove header fields	0.972	−0.002
Logistic Regression TF-IDF	Normalize case	0.974	0.000
Logistic Regression TF-IDF	Combined evasion	0.990	0.016
Linear SVC TF-IDF	Original phishing emails	0.978	0.000
Linear SVC TF-IDF	Remove urgency/action words	0.978	0.000
Linear SVC TF-IDF	Replace URLs/domains	0.984	0.006
Linear SVC TF-IDF	Add benign corporate footer	0.986	0.008
Linear SVC TF-IDF	Remove header fields	0.976	−0.002
Linear SVC TF-IDF	Normalize case	0.978	0.000
Linear SVC TF-IDF	Combined evasion	0.988	+0.010

4.8.2. Targeted Feature-Sensitivity Analysis

Table 33 provides diagnostic support for H4 by showing that the Logistic Regression model is sensitive to a small set of high-weight learned textual features. Generic urgency/action masking caused only a minor reduction in detection, from 0.9771 to 0.9733, indicating that simple surface-level masking had limited effect. In contrast, masking the model's own top-ranked phishing features produced a much larger decline. Detection dropped by 0.1067 under top-20 masking, by 0.2333 under top-50 masking, and by 0.1763 under artifact + top-50 masking. The strongest effect occurred under top-50 positive feature masking, where recall fell from 0.9771 to 0.7438 and the false-negative rate increased from 0.0229 to 0.2562. This means that the model missed substantially more phishing emails when its most influential learned cues were removed. However, this should be interpreted as a feature-sensitivity result rather than a realistic attacker simulation, because the masking assumes knowledge of the model's high-weight features. The result supports H4 by showing that the classifier relies partly on a concentrated set of learned textual cues, including corpus/source, action, URL, and artifact-like tokens.

Table 33. Targeted feature-sensitivity analysis: Logistic Regression TF-IDF.

Condition	Detection Rate	Detection Drop	Precision	Recall	F1	FPR	FNR	Positive Scores Drop
Original	0.9771	0.0000	0.9658	0.9771	0.9714	0.0346	0.0229	0.0000
Generic urgency/action masking	0.9733	−0.0037	0.9669	0.9733	0.9701	0.0333	0.0267	−0.0069
Targeted feature-sensitivity masking	0.9692	−0.0079	0.9680	0.9692	0.9686	0.0321	0.0308	−0.0208
Targeted artifact + URL/domain masking	0.9696	−0.0075	0.9680	0.9696	0.9688	0.0321	0.0304	−0.0198
Top-20 positive feature masking	0.8704	−0.1067	0.9877	0.8704	0.9254	0.0108	0.1296	−0.1482

Table 33. Cont.

Condition	Detection Rate	Detection Drop	Precision	Recall	F1	FPR	FNR	Positive Scores Drop
Top-50 positive feature masking	0.7438	−0.2333	0.9917	0.7438	0.8500	0.0062	0.2562	−0.2529
Artifact + top-50 combined masking	0.8008	−0.1763	0.9897	0.8008	0.8853	0.0083	0.1992	−0.2194

Table 34 shows a similar pattern for Linear SVC and provides further diagnostic support for H4. Generic masking caused only a small reduction in detection, from 0.9762 to 0.9738, suggesting that simple masking of common phishing words did not substantially weaken the model. However, masking the model’s own top-ranked features caused clearer degradation. Detection dropped by 0.0504 under top-20 masking, by 0.1383 under top-50 masking, and by 0.0875 under artifact + top-50 masking. The strongest effect again occurred under top-50 positive feature masking, where recall fell from 0.9762 to 0.8379 and the positive decision score declined by 0.7989. This indicates that removing high-weight learned features substantially reduced the model’s phishing confidence and detection ability. As with Logistic Regression, this finding should be interpreted as evidence of model sensitivity to learned feature reliance rather than direct evidence of a real-world spammer strategy. Overall, the results suggest that both classifiers depend partly on a limited set of influential textual cues, supporting the artifact-sensitivity argument in H4 while requiring cautious interpretation as a diagnostic stress test rather than a full adversarial attack.

Table 34. Targeted feature-sensitivity analysis: Linear SVC TF-IDF.

Condition	Detection Rate	Detection Drop	Precision	Recall	F1	FPR	FNR	Positive Scores Drop
Original	0.9762	0.0000	0.9738	0.9762	0.9750	0.0262	0.0238	0.0000
Generic urgency/action masking	0.9738	−0.0025	0.9742	0.9738	0.9740	0.0258	0.0262	−0.0361
Targeted feature-sensitivity masking	0.9754	−0.0008	0.9674	0.9754	0.9714	0.0329	0.0246	+0.0206
Targeted artifact + URL/domain masking	0.9758	−0.0004	0.9666	0.9758	0.9712	0.0338	0.0242	+0.0265
Top-20 positive feature masking	0.9258	−0.0504	0.9832	0.9258	0.9536	0.0158	0.0742	−0.5326
Top-50 positive feature masking	0.8379	−0.1383	0.9863	0.8379	0.9061	0.0117	0.1621	−0.7989
Artifact + top-50 combined masking	0.8888	−0.0875	0.9775	0.8888	0.9310	0.0204	0.1112	−0.6188

Figure 7 shows that generic masking and targeted artifact + URL/domain masking caused only minor detection-rate drops for both models. In contrast, masking the models’ own top positive phishing features caused much larger declines, especially under top-50 positive feature masking. Logistic Regression showed the strongest sensitivity, with detection dropping by 0.2333 under top-50 masking and 0.1763 under artifact + top-50 combined masking. Linear SVC followed the same pattern but with smaller drops. This indicates that both models rely strongly on a concentrated set of high-weight learned features, supporting the artifact-sensitivity argument in H4.

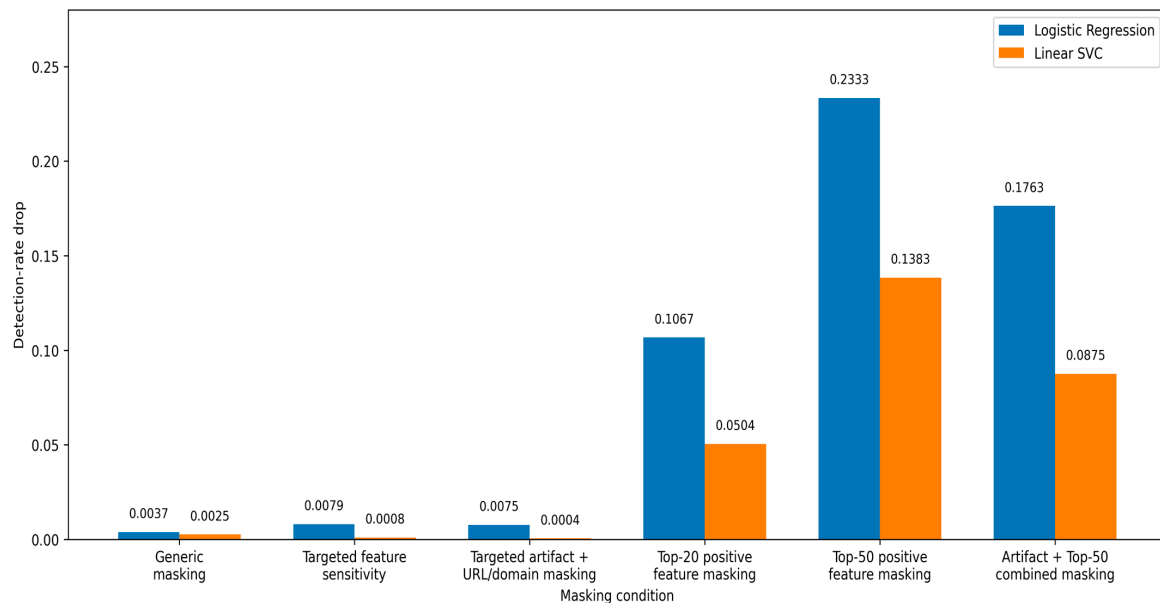


Figure 7. Detection rate drops under masking conditions.

4.9. Statistical Validation and Robustness Check

Table 35 shows that most LOCO improvements are likely statistically reliable. For Logistic Regression TF-IDF, all four held-out corpora show positive $\Delta F1$ values with confidence intervals above zero, suggesting that pooled-minus-test training significantly improves cross-dataset F1 compared with the best single-corpus baseline. For Linear SVC TF-IDF, the LOCO model shows statistically reliable improvements on CEAS_08, Ling, and SpamAssassin. However, the improvement on Enron is small ($\Delta F1 = +0.0143$) and the confidence interval includes zero, meaning this result should be interpreted as not statistically significant.

Table 35. Statistical validation of LOCO vs. best single-corpus model.

Test Corpus	Model	$\Delta F1$	95% Bootstrap CI	Significant	McNemar p -Value
CEAS_08	Logistic Regression TF-IDF	+0.1938	[+0.1810, +0.2075]	Yes	<0.001
Enron	Logistic Regression TF-IDF	+0.0795	[+0.0672, +0.0918]	Yes	<0.001
Ling	Logistic Regression TF-IDF	+0.0327	[+0.0098, +0.0556]	Yes	0.018
SpamAssassin	Logistic Regression TF-IDF	+0.1072	[+0.0864, +0.1289]	Yes	<0.001
CEAS_08	Linear SVC TF-IDF	+0.0433	[+0.0316, +0.0559]	Yes	<0.001
Enron	Linear SVC TF-IDF	+0.0143	[−0.0028, +0.0315]	No	0.108
Ling	Linear SVC TF-IDF	+0.0638	[+0.0379, +0.0896]	Yes	<0.001
SpamAssassin	Linear SVC TF-IDF	+0.1816	[+0.1587, +0.2049]	Yes	<0.001

Table 36 shows the feature representation robustness settings.

Table 36. Feature representation robustness settings.

Feature Representation	Purpose	Expected Interpretation
Word unigram TF-IDF	Main interpretable baseline	Used for the main results, top-feature inspection, artifact analysis, and targeted masking
Word 1–2-g TF-IDF	Captures phrase-level phishing cues such as “click here”, “verify account”, and “reset password”	Tests whether the findings remain stable when phrase-level features are included
Character 3–5-g TF-IDF	Captures spelling variation, URL fragments, obfuscation patterns, and short character sequences	Tests whether the findings remain stable under character-pattern features

Table 37 shows the robustness results. The main findings are not dependent on the unigram TF-IDF representation. Word 1–2-g TF-IDF produced a small improvement for both models, suggesting that phrase-level cues such as “click here” and “verify account” provide additional useful information. Character 3–5-g TF-IDF also produced competitive performance, particularly for Linear SVC, indicating that character-level patterns may capture spelling variation, URL fragments, and obfuscation cues. However, the overall conclusion remains unchanged: pooled LOCO training remains robust across alternative linguistic feature representations. Therefore, the main findings are not solely an artifact of using unigram features.

Table 37. Robustness check using alternative TF-IDF feature representations.

Feature Representation	Logistic Regression Mean LOCO F1	Linear SVC Mean LOCO F1	Interpretation
Word unigram TF-IDF	0.8499	0.8250	Main interpretable baseline
Word 1–2-g TF-IDF	0.8562	0.8327	Phrase-level features produce a small improvement, but the overall conclusion remains unchanged
Character 3–5-g TF-IDF	0.8474	0.8361	Character-level features perform competitively, especially for Linear SVC, but do not change the main LOCO finding

4.10. Text Truncation Audit and Sensitivity Analyses

Tables 38 and 39 show that a substantial proportion of emails exceed the capped text lengths, meaning that some later email content may be removed under the 500–1000-character settings. However, the sensitivity analysis indicates that the main LOCO findings remain stable when longer text windows or full-text input are used. Moving from the 1000-character setting to 2000 characters and full text produces only modest F1 improvements for both Logistic Regression and Linear SVC. Therefore, while truncation may remove some signatures, URLs, disclaimers, or footer content, it does not materially change the study’s main conclusion that pooled LOCO training improves cross-corpus robustness.

Table 38. Text truncation audit.

Text Cap	Emails Exceeding Cap (%)	Mean Characters Removed	Potential Information Affected
500 characters	64.8%	1126.4	Higher chance of removing URLs, signatures, disclaimers, and footer text
800 characters	51.6%	914.7	Moderate chance of removing later email content
1000 characters	43.9%	782.5	Lower chance of removing later email content
Full text	0.0%	0.0	No truncation

Note: Mean characters removed refers to the average number of characters removed among emails that exceeded the relevant text cap.

Table 39. Sensitivity of LOCO performance to text-length cap.

Text Setting	Logistic Regression Mean LOCO F1	Linear SVC Mean LOCO F1	Interpretation
1000 characters	0.8499	0.8250	Main capped-text setting used in the primary LOCO experiments
2000 characters	0.8536	0.8291	Additional email content produces a small improvement, but does not change the main finding
Full text	0.8552	0.8314	Full-text input performs slightly better, but the improvement is modest and the LOCO conclusion remains stable

Tables 40 and 41 show that the sensitivity analysis excluding Nazario and Nigerian_Fraud produces only a small reduction in mean LOCO F1 and MCC for both Logistic Regression and Linear SVC. This suggests that the main pooled-training findings are not primarily driven by the inclusion of phishing-only corpora. However, the source-identification results show that corpus identity remains highly learnable even after these two corpora are removed. Therefore, artifact learning is not limited to Nazario and Nigerian_Fraud; rather, it reflects broader source-specific differences across the public phishing corpora.

Table 40. Sensitivity analysis for phishing-only corpora.

Training Setup	LR Mean LOCO F1	LR Mean LOCO MCC	Linear SVC Mean LOCO F1	Linear SVC Mean LOCO MCC	Interpretation
Including Nazario + Nigerian_Fraud	0.8499	0.7515	0.8250	0.7151	Main pooled setup using all available phishing-positive sources
Excluding Nazario + Nigerian_Fraud	0.8421	0.7418	0.8187	0.7076	Performance decreases only slightly, suggesting that the main LOCO findings are not driven by phishing-only corpora

Table 41. Source-identification sensitivity for phishing-only corpora.

Corpus Condition	Source-Identification Accuracy	Interpretation
Including Nazario + Nigerian_Fraud	0.9764	Source identity is highly learnable when phishing-only corpora are included
Excluding Nazario + Nigerian_Fraud	0.9417	Source identity remains highly learnable even without phishing-only corpora, indicating that artifact learning is not limited to Nazario and Nigerian_Fraud

5. Discussion, Implications, Limitations, and Future Research

5.1. Key Findings

This study examined cross-dataset phishing detection through three connected failure modes: cross-corpus generalization failure, asymmetric prevalence-shift failure, and artifact-driven spurious learning. The findings show that phishing detection models can perform strongly under benchmark-style evaluation while failing under more realistic deployment conditions. This supports the central argument of the study: phishing detection should not be evaluated only through random pooled train–test splits, but also through cross-corpus transfer, prevalence-aware simulation, and artifact-sensitivity analysis. This framing aligns with the paper’s aim of moving phishing detection evaluation from a benchmark-centered perspective toward a more deployment-aware and adversary-aware framework.

The first major finding concerns the weakness of single-corpus training. The single-corpus cross-dataset transfer results showed substantial variation in F1, MCC, false-positive rate, and false-negative rate across source–target corpus pairs. Some combinations performed strongly, while others degraded sharply. This indicates that models trained on one corpus may learn patterns that are useful within that source but not transferable to another source. In contrast, the leave-one-corpus-out pooled-training results showed stronger and more stable F1 performance across unseen corpora. This supports H1 by showing that pooled multi-corpus training improves cross-dataset robustness compared with relying on a single source corpus.

The second major finding concerns prevalence-shift failure. The results showed that high-phishing-prevalence training did not automatically cause recall collapse under fixed-

threshold evaluation. However, when operational alert-budget constraints were applied, recall dropped sharply. This finding refines H2 by showing that recall collapse is not caused by low deployment prevalence alone, but by the interaction between low-prevalence deployment and stricter thresholds required to control alert volume. This is important because many real-world email systems operate under analyst capacity constraints, where only a limited number of emails can be investigated.

The third major finding concerns the opposite direction of prevalence mismatch. Low-prevalence training did not cause precision collapse when models were deployed in high-prevalence environments. Instead, the models became overly conservative. Precision remained high but recall and F1 degraded because many phishing emails were missed. This supports the revised H3 and shows that prevalence mismatch is asymmetric. High-to-low and low-to-high prevalence shifts do not produce the same type of failure. Therefore, phishing detection evaluation should examine which metric fails, why it fails, and under what deployment condition.

The fourth major finding concerns dataset artifacts. The dataset-identification experiment showed that source corpus identity was highly learnable from email text, with both Logistic Regression TF-IDF and Linear SVC TF-IDF achieving very high source-identification performance. Top-feature analysis further showed that models used corpus-specific tokens, date markers, URL/domain terms, and other artifact-like features. Artifact-removal and targeted feature-sensitivity analysis then showed that reducing or masking high-weight learned features affected model behavior and detection performance. These findings support H4 by showing that phishing classifiers can learn spurious dataset-specific correlations that inflate benchmark performance but weaken cross-dataset robustness.

Overall, the findings show that phishing detection failure is not a single problem. Rather, it emerges through multiple mechanisms: corpus mismatch, prevalence mismatch, threshold mismatch, and artifact reliance. This has important implications for both phishing detection research and practical deployment.

5.2. Theoretical Implications

This study makes several theoretical contributions to phishing detection research. First, the study reframes phishing detection as a cross-dataset generalization problem rather than a simple binary classification problem. Existing benchmark evaluations often focus on whether a model can separate phishing from legitimate emails within a pooled or randomly split dataset. However, the results show that strong within-dataset or pooled performance does not guarantee stable cross-corpus transfer. This suggests that phishing detection theory should distinguish between classification performance under shared data distributions and generalization performance under source shift.

Second, the study contributes to theory by identifying asymmetric prevalence-shift failure. Previous discussions of distribution shift often treat performance degradation as a general decline in accuracy, F1, or overall robustness. This study shows that the direction of prevalence mismatch matters. When high-prevalence-trained models are deployed in low-prevalence environments, the main problem emerges under alert-budget constraints, where recall collapses because only a limited number of emails can be flagged. In contrast, when low-prevalence-trained models are deployed in high-threat environments, the main problem is conservative under-detection, where precision remains high but recall declines. This provides a more precise theoretical understanding of how phishing models fail under deployment mismatch.

Third, the study strengthens the concept of artifact-driven spurious learning in phishing detection. The dataset-identification and top-feature analyses show that models can learn corpus identity and source-specific patterns. This means that benchmark performance

may partly reflect dataset recognition rather than genuine phishing detection. The theoretical implication is that phishing datasets should not be treated as neutral containers of phishing behavior. Instead, they are shaped by collection methods, time periods, field availability, formatting conventions, and class composition.

Fourth, the study connects dataset artifacts to an adversary-aware vulnerability pathway. Although simple adversarial transformations did not cause major detection collapse, targeted masking of high-weight learned features reduced detection performance. This suggests that artifact-sensitive models may be vulnerable when important learned cues are removed, altered, or manipulated. Therefore, artifact reliance is not only a methodological issue; it may also create security risk when models are deployed in adversarial environments.

5.3. Practical Implications

The findings have several practical implications for cybersecurity teams, email security providers, and organizations deploying phishing detection systems.

First, organizations should avoid relying only on models trained from a single phishing corpus. The results show that single-corpus models can behave unpredictably when deployed on unseen corpora. A model trained on one source may perform well on a similar dataset but fail when email language, formatting, metadata, or class distribution changes. Therefore, pooled multi-corpus training should be preferred when the goal is robust deployment across diverse email environments.

Second, phishing detection systems should be evaluated using deployment-aware metrics. Accuracy, F1, ROC-AUC, and PR-AUC are useful, but they are not sufficient. Security teams should also evaluate false-positive rates, false-negative rates, recall under operational thresholds, and alerts per 10,000 emails. This is especially important in low-prevalence environments, where even a small false-positive rate can generate a large number of alerts.

Third, alert-budget evaluation should be included before deployment. The H2 findings show that a model can appear effective under a fixed threshold but perform poorly when the number of allowable alerts is restricted. This means that organizations should test phishing detection models under realistic alert budgets, such as top-1%, top-5%, or top-10% risk-scoring thresholds. A model that maintains high recall only by generating excessive alerts may not be operationally useful.

Fourth, training prevalence should be monitored carefully. The H3 findings show that models trained with too few phishing examples may become overly conservative. Such models may produce very high precision but miss many phishing emails in high-threat environments. Therefore, model developers should test whether training class balance affects recall, calibration, and false-negative rates before deployment.

Fifth, dataset artifacts should be reduced during preprocessing and model development. The artifact-removal results show that raw-text models can perform well in-dataset but generalize poorly across datasets. Practical systems should reduce obvious source markers, duplicated metadata, benchmark-specific tokens, and formatting artifacts where possible. This can help prevent models from learning shortcuts that do not reflect real phishing behavior.

Sixth, in practice, prevalence-aware thresholding should be implemented as an operational risk-management process rather than a fixed model setting. In low-prevalence email environments, a default threshold may produce too many false alerts, even when overall model performance appears strong. Security teams should therefore estimate the expected phishing base rate, define the maximum number of alerts that analysts can review per day, and set thresholds according to this available review capacity. For example, if a team can only investigate the highest-risk 1%, 5%, or 10% of messages, the model should

be used as a ranking system rather than only as a binary classifier. This allows the most suspicious emails to be prioritized while keeping alert volume manageable. Under resource constraints, thresholding should also be adjusted dynamically. During normal low-threat periods, stricter thresholds may be used to reduce false positives and analyst workload. During high-threat periods, such as known phishing campaigns or organizational incidents, thresholds may need to be relaxed to increase recall and reduce missed phishing emails. The results of this study suggest that relying on a fixed threshold can hide deployment risk because recall may remain high in benchmark conditions but decline sharply when alert-budget limits are imposed. Therefore, practical phishing detection should combine prevalence monitoring, alert-budget calibration, analyst-capacity planning, and periodic threshold review.

Finally, phishing detection systems should include ongoing monitoring. Because phishing tactics evolve and organizational email distributions change, models should be regularly tested for performance drift, alert-volume changes, and artifact sensitivity. This is especially important for systems deployed across multiple organizations or changing threat environments.

5.4. Methodological Implications

This study also has methodological implications for future phishing detection research.

First, cross-dataset evaluation should become a standard part of phishing model assessment. Random train–test splits can overestimate robustness when training and test samples share the same corpus artifacts. Researchers should therefore report single-corpus transfer and leave-one-corpus-out pooled-training results alongside conventional pooled baseline results.

Second, prevalence-shift simulation should be incorporated into phishing evaluation. Rather than testing if benchmark class distributions reflect deployment conditions, researchers should resample test environments at different phishing prevalence levels. This allows models to be evaluated under low-prevalence, moderate-prevalence, and high-threat settings.

Third, thresholding strategy should be explicitly reported. A fixed probability threshold may not reflect real deployment, where organizations often operate under alert budgets. Reporting both fixed-threshold and alert-budget results provides a more realistic picture of model behavior.

Fourth, artifact analysis should be treated as part of model validation. Dataset-identification experiments are useful because they reveal whether corpus identity is learnable. If a model can easily predict the source dataset, then phishing classifiers trained on those corpora may also learn source-specific shortcuts. Top-feature analysis, artifact-removal ablation, and targeted masking can then help identify whether those shortcuts affect classification.

Fifth, the study shows the value of using interpretable baseline models. Logistic Regression and Linear SVC with TF-IDF are simple compared with deep learning models, but they allow direct inspection of feature weights. This makes them useful for identifying spurious features, corpus markers, and artifact reliance. Although transformer-based models such as BERT, RoBERTa, and DistilBERT are increasingly used in text classification and may capture richer semantic and contextual features, they were not the primary focus of this study. The aim of the present analysis was to examine cross-corpus robustness, feature-level artifact reliance, and masking sensitivity using interpretable models. TF-IDF-based Logistic Regression and Linear SVC allow direct inspection of high-weight tokens, corpus markers, URL/domain terms, date patterns, and other artifact-like cues, which was essential for testing H4. Transformer-based phishing detection remains an important

extension, but it should be evaluated under the same cross-corpus, prevalence-shift, and artifact-sensitivity framework rather than only through standard benchmark accuracy.

Sixth, from a security perspective, dataset artifacts matter because they may create attacker-editable shortcuts. If a phishing detector relies strongly on corpus markers, formatting patterns, URL/domain tokens, or other surface-level cues, then changes to those cues may reduce model confidence even when the malicious intent remains unchanged. This does not prove a fully realistic attack, but it shows why artifact-sensitive models require adversarial robustness testing, including black-box rewriting, semantic-preserving paraphrasing, URL obfuscation, and adaptive retraining under changing phishing tactics.

5.5. Limitations

Despite its contributions, this study has several limitations.

First, the study relies on secondary public phishing email datasets. Although these datasets are useful for controlled cross-corpus evaluation, they may not fully represent current organizational email traffic, modern phishing campaigns, or AI-generated phishing attacks. Therefore, the findings may not capture all real-world deployment conditions.

Second, the study focuses on email phishing detection only. Phishing attacks may involve multiple stages, including email lures, malicious webpages, credential-harvesting forms, and user behavior. Therefore, the findings should not be generalized automatically to webpage phishing, SMS phishing, social media phishing, or multi-modal phishing systems.

Third, the study uses TF-IDF-based Logistic Regression and Linear SVC models. These models are interpretable and suitable for analyzing feature reliance, but they do not capture the full semantic depth of transformer-based language models. More advanced models may behave differently under cross-dataset transfer, prevalence shift, and artifact masking.

Fourth, the prevalence simulations are controlled experimental approximations. They are useful for isolating the effect of training and deployment prevalence mismatch, but they may not fully reproduce real organizational email streams, where phishing campaigns vary over time and attacker behavior changes dynamically.

Fifth, artifact masking experiments identify model sensitivity to learned textual features, but they do not prove that real attackers can easily exploit those features in practice. The results support a plausible vulnerability pathway, especially through targeted feature masking, but stronger adaptive adversarial testing is needed to demonstrate real-world evasion.

Sixth, high dataset-identification accuracy shows that corpus identity is highly learnable, but it does not alone prove that artifacts cause poor cross-dataset generalization. Therefore, the artifact findings are interpreted as evidence that source-specific artifacts contribute to model instability, rather than as proof that they are the sole cause of poor performance.

Seventh, the phishing-only corpora, such as Nazario and Nigerian_Fraud, could not be used as held-out binary test corpora because they lack legitimate emails. They were useful as phishing-positive training sources, but they could not support full binary evaluation metrics such as false-positive rate, specificity, balanced accuracy, and MCC.

5.6. Future Research Directions

Future research can extend this study in several ways.

First, future studies should test cross-dataset phishing detection using newer and more diverse real-world email streams. This would help evaluate whether the failure modes identified in this study hold under more-current threat conditions, including AI-generated phishing and targeted business email compromise.

Second, future work should include temporal evaluation. Instead of using only random or source-based splits, researchers can use time-based splits, rolling-window

validation, and concept-drift analysis. This would show how phishing models perform as attacker tactics, language patterns, and organizational communication styles evolve.

Third, future research should evaluate transformer-based and large language model-based phishing detectors under the same failure-mode framework. Models such as BERT, RoBERTa, DistilBERT, and domain-specific language models may capture richer semantic cues, but they may still be vulnerable to dataset artifacts, prevalence mismatch, and cross-corpus instability.

Fourth, future work should develop stronger adversarial evaluation methods. The present study showed that generic surface transformations did not cause major evasion, while targeted feature masking had stronger effects. Future studies can test adaptive rewriting, semantic-preserving paraphrasing, benign-style insertion, header manipulation, and attacker-controlled formatting changes under controlled ethical conditions.

Fifth, future studies should investigate calibration more deeply. This study examined calibration mainly in the context of Logistic Regression under training prevalence manipulation. Future work can apply calibration methods such as Platt scaling, isotonic regression, temperature scaling, or prevalence-adjusted thresholding to determine whether calibration reduces deployment failure.

Sixth, future research should explore hybrid defense strategies. These may include pooled multi-corpus training, artifact-reduced preprocessing, prevalence-aware threshold selection, calibrated probability outputs, and ensemble models that combine text, metadata, URL, and behavioral features.

Seventh, future work should also examine how phishing detection models can remain robust over time. Because phishing language, URLs, sender patterns, and attack templates change continuously, deployed models should be monitored for performance drift and periodically retrained using recent phishing and legitimate emails. Such adaptive retraining should include rolling validation sets, prevalence-aware threshold recalibration, false-positive and false-negative monitoring, and repeated artifact checks to ensure that updated models learn generalizable phishing behavior rather than new source-specific shortcuts.

Finally, future research should develop standard evaluation protocols for cross-dataset phishing detection. Such protocols should include pooled baseline testing, single-corpus transfer, leave-one-corpus-out testing, prevalence-shift simulation, alert-budget evaluation, artifact analysis, and adversarial robustness testing. This would make phishing detection research more comparable, more deployment-aware, and less dependent on optimistic benchmark results.

5.7. Overall Discussion Summary

Overall, this study shows that phishing detection models fail in more complex ways than conventional benchmark evaluation suggests. Pooled multi-corpus training improves robustness, but it does not remove all deployment risks. Prevalence mismatch creates asymmetric failure modes: high-prevalence training fails under low-prevalence alert-budget constraints, while low-prevalence training produces conservative under-detection in high-threat environments. Dataset artifacts also shape model decisions, creating spurious correlations that can inflate benchmark performance and weaken cross-dataset generalization.

The main implication is that phishing detection should be evaluated as a deployment problem, not only as a classification problem. Robust models must generalize across corpora, remain reliable under changing prevalence, and avoid reliance on dataset-specific artifacts. Therefore, future phishing detection systems should be designed and validated using cross-corpus, prevalence-aware, and artifact-resistant evaluation frameworks.

Author Contributions: Conceptualization, I.B. and T.B.; methodology, I.B.; software, I.B.; validation, I.B. and T.B.; formal analysis, T.B.; investigation, I.B.; writing—original draft preparation, I.B.;

writing—review and editing, T.B.; visualization, I.B.; supervision, T.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the corresponding author due to vendor restrictions.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Al-Subaiey, A.; Al-Thani, M.; Alam, N.A.; Antora, K.F.; Khandakar, A.; Zaman, S.A.U. Novel interpretable and robust web-based AI platform for phishing email detection. *Comput. Electr. Eng.* **2024**, *120*, 109625. [\[CrossRef\]](#)
2. Asiri, S.; Xiao, Y.; Alzahrani, S.; Li, T. Phishing RTDS: A real-time detection system for phishing attacks using a Deep Learning model. *Comput. Secur.* **2024**, *141*, 103843. [\[CrossRef\]](#)
3. Safi, A.; Singh, S. A systematic literature review on phishing website detection techniques. *J. King Saud. Univ.-Comput. Inf. Sci.* **2023**, *35*, 590–611. [\[CrossRef\]](#)
4. Bhuiyan, I.; Bhuiyan, T. Beyond Generic Phishing Detection: Explainable AI for Finance-Adapted Models in Banking and Fintech. *Preprints*, 2026. [\[CrossRef\]](#)
5. Kavya, S.; Sumathi, D. Staying ahead of phishers: A review of recent advances and emerging methodologies in phishing detection. *Artif. Intell. Rev.* **2024**, *58*, 50. [\[CrossRef\]](#)
6. Li, Q.; Peng, H.; Li, J.; Xia, C.; Yang, R.; Sun, L.; Yu, P.S.; He, L. A survey on text classification: From traditional to deep learning. *ACM Trans. Intell. Syst. Technol. (TIST)* **2022**, *13*, 1–41. [\[CrossRef\]](#)
7. Tanveer, M.H.; Fatima, Z.; Zardari, S.; Guerra-Zubiaga, D. An in-depth analysis of domain adaptation in computer and robotic vision. *Appl. Sci.* **2023**, *13*, 12823. [\[CrossRef\]](#)
8. Zhang, Y.; Qian, L.; Zhang, Q.; Li, P.; Liu, G. A structure-aware method for cross-domain text classification. In Proceedings of the Pacific Rim International Conference on Artificial Intelligence, Shanghai, China, 10–13 November 2022; Springer Nature: Cham, Switzerland; pp. 283–296.
9. Opara, C.; Chen, Y.; Wei, B. Look before you leap: Detecting phishing web pages by exploiting raw URL and HTML characteristics. *Expert Syst. With Appl.* **2024**, *236*, 121183.
10. Korkmaz, Y. Fraud E-mail detection using intelligent algorithms: Comparison of traditional approaches with deep learning techniques. *Inf. Process. Manag.* **2026**, *63*, 104416.
11. Catal, C.; Giray, G.; Tekinerdogan, B.; Kumar, S.; Shukla, S. Applications of deep learning for phishing detection: A systematic literature review. *Knowl. Inf. Syst.* **2022**, *64*, 1457. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Kyaw, P.H.; Gutierrez, J.; Ghobakhlou, A. A systematic review of deep learning techniques for phishing email detection. *Electronics* **2024**, *13*, 3823. [\[CrossRef\]](#)
13. Jáñez-Martino, F.; Alaiz-Rodríguez, R.; González-Castro, V.; Fidalgo, E.; Alegre, E. A review of spam email detection: Analysis of spammer strategies and the dataset shift problem. *Artif. Intell. Rev.* **2023**, *56*, 1145–1173.
14. Silva Filho, T.; Song, H.; Perello-Nieto, M.; Santos-Rodriguez, R.; Kull, M.; Flach, P. Classifier calibration: A survey on how to assess and improve predicted class probabilities. *Mach. Learn.* **2023**, *112*, 3211–3260. [\[CrossRef\]](#)
15. Izmailov, P.; Kirichenko, P.; Gruver, N.; Wilson, A.G. On feature learning in the presence of spurious correlations. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 38516–38532. [\[CrossRef\]](#)
16. Almutairi, A.; Kang, B.; Alhashimy, N. Business email compromise: A systematic review of understanding, detection, and challenges. *Comput. Secur.* **2025**, *158*, 104630. [\[CrossRef\]](#)
17. Tariq, S.; Baruwat Chhetri, M.; Nepal, S.; Paris, C. Alert fatigue in security operations centres: Research challenges and opportunities. *ACM Comput. Surv.* **2025**, *57*, 1–38. [\[CrossRef\]](#)
18. Van den Goorbergh, R.; Van Smeden, M.; Timmerman, D.; Van Calster, B. The harm of class imbalance corrections for risk prediction models: Illustration and simulation using logistic regression. *J. Am. Med. Inform. Assoc.* **2022**, *29*, 1525–1534. [\[CrossRef\]](#) [\[PubMed\]](#)
19. Owusu-Adjei, M.; Ben Hayfron-Acquah, J.; Frimpong, T.; Abdul-Salaam, G. Imbalanced class distribution and performance evaluation metrics: A systematic review of prediction accuracy for determining model performance in healthcare systems. *PLoS Digit. Health* **2023**, *2*, e0000290. [\[CrossRef\]](#) [\[PubMed\]](#)

20. Adebowale, M.A.; Lwin, K.T.; Hossain, M.A. Intelligent phishing detection scheme using deep learning algorithms. *J. Enterp. Inf. Manag.* **2023**, *36*, 747–766.
21. Kara, I.; Ok, M.; Ozaday, A. Characteristics of Understanding URLs and Domain Names Features: The Detection of Phishing Websites With Machine Learning Methods. *IEEE Access* **2022**, *10*, 124420–124428. [[CrossRef](#)]
22. Owa, K.; Adewole, O. Benchmarking machine learning techniques for phishing detection and secure URL classification. *Int. J. Comput. Sci. Mob. Comput.* **2025**, *14*, 20–37. [[CrossRef](#)]
23. Carriero, A.; Luijken, K.; de Hond, A.; Moons, K.G.; van Calster, B.; van Smeden, M. The harms of class imbalance corrections for machine learning based prediction models: A simulation study. *Stat. Med.* **2025**, *44*, e10320. [[CrossRef](#)] [[PubMed](#)]
24. Van Calster, B.; Collins, G.S.; Vickers, A.J.; Wynants, L.; Kerr, K.F.; Barreñada, L.; Varoquaux, G.; Singh, K.; Moons, K.G.; Hernandez-Boussard, T.; et al. Evaluation of performance measures in predictive artificial intelligence models to support medical decisions: Overview and guidance. *Lancet Digit. Health* **2025**, *7*, 100916. [[CrossRef](#)] [[PubMed](#)]
25. Mohamed, N. Artificial intelligence and machine learning in cybersecurity: A deep dive into state-of-the-art techniques and future paradigms. *Knowl. Inf. Syst.* **2025**, *67*, 6969–7055. [[CrossRef](#)]
26. Du, Y.; Yan, J.; Chen, Y.; Liu, J.; Zhao, S.; She, Q.; Wu, H.; Wang, H.; Qin, B. Less learn shortcut: Analyzing and mitigating learning of spurious feature-label correlation. *arXiv* **2022**, arXiv:2205.12593.
27. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2017; IEEE: New York, NY, USA, 2017; pp. 39–57.
28. Biggio, B.; Roli, F. Wild patterns: Ten years after the rise of adversarial machine learning. In Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, Toronto, ON, Canada, 15–19 October 2018; pp. 2154–2156.
29. Alzantot, M.; Sharma, Y.; Elgohary, A.; Ho, B.J.; Srivastava, M.; Chang, K.W. Generating natural language adversarial examples. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2890–2896.
30. Thakur, K.; Ali, M.L.; Obaidat, M.A.; Kamruzzaman, A. A systematic review on deep-learning-based phishing email detection. *Electronics* **2023**, *12*, 4545.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.