

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

Mohammed A Almjbray^{1*}, Issa M Almagrhe

¹ Higher Institute of Engineering Technologies, Al-Majouri – Benghazi, Libya

² Higher Institute of Engineering Technologies, Al-Majouri – Benghazi, Libya

*Corresponding author: mohammedalmajbray97@gmail.com

التصنيف متعدد الفئات لرسائل البريد الإلكتروني باستخدام التقييم الواعي

بتسرب البيانات والتعلم الآلي

محمد المجبري^{1*}، عيسى المقرحي²

¹ المعهد العالي للتقنيات الهندسية الماجوري – بنغازي ، ليبيا

² المعهد العالي للتقنيات الهندسية الماجوري – بنغازي ، ليبيا

Received: 15/8/2026 Revised: 30/8/2026 Accepted: 05/9/2026 Publication: 29/9/2026

Abstract: Reliable evaluation is important in multi-class email classification because aggregated text datasets may contain duplicated messages or overlapping content that can affect separation between model-development and evaluation data. This study presents a leakage-aware framework for classifying emails into Ham, Phish, and Spam. The framework integrates data cleaning, duplicate auditing, removal of conflicting duplicate groups, group-aware partitioning, training-only feature construction, stratified group cross-validation, locked-test evaluation, error analysis, and bootstrap confidence intervals. The downloaded dataset contained 365,448 records; after cleaning and removing conflicting groups, 365,403 remained. A five-fold StratifiedGroupKFold procedure produced a locked test set of 73,038 messages with zero duplicate-group overlap. Word-level TF-IDF with LinearSVC was used as the primary classifier, while a complementary H2O AutoML branch used 128-component SVD. LinearSVC achieved 98.00% accuracy and 97.32% macro F1 on the locked test set. The findings emphasize the importance of controlling duplicate structure and information flow when evaluating multi-class email classifiers.

Keywords: Email classification, spam detection, phishing detection, data leakage, duplicate-aware evaluation, TF-IDF, LinearSVC, automated machine learning.

المخلص: يُعد التقييم الموثوق مهماً في تصنيف رسائل البريد الإلكتروني متعددة الفئات، خصوصاً عند استخدام مجموعات بيانات مجمعة قد تحتوي على رسائل مكررة أو محتوى متطابق. تقدم هذه الدراسة إطاراً واعياً بتسرب البيانات لتصنيف الرسائل إلى ثلاث فئات Ham و Phish و Spam. يدمج الإطار تنظيف البيانات، وتدقيق التكرارات، وإزالة المجموعات ذات التسميات المتعارضة، والتقسيم مع مراعاة المجموعات، وبناء الخصائص من بيانات التدريب فقط، والتحقق المتقاطع الطبقي، والتقييم باستخدام مجموعة اختبار مغلقة، وتحليل الأخطاء، وفواصل الثقة بطريقة Bootstrap. احتوت البيانات المحملة على 365,448 سجلاً، وبقي 365,403 سجلاً بعد التنظيف وإزالة المجموعات المتعارضة. أنتج إجراء StratifiedGroupKFold الطيات مجموعة اختبار تضم 73,038 رسالة دون تداخل بين مجموعات التكرارات. استُخدم TF-IDF على مستوى الكلمات مع LinearSVC كنموذج أساسي، إلى جانب فرع مقارن باستخدام H2O AutoML وتمثيل SVD من 128 بُعداً. حقق LinearSVC دقة 98.00% و Macro F1 بنسبة 97.32% على مجموعة الاختبار. تؤكد النتائج أهمية التحكم في التكرارات وتدفق المعلومات عند تقييم مصنفات البريد متعددة الفئات.

الكلمات المفتاحية: تصنيف البريد الإلكتروني، الرسائل المزعجة، التصيد الاحتيالي، تسرب البيانات، التكرارات، TF-IDF، LinearSVC، التعلم الآلي.

1. INTRODUCTION

Electronic mail remains an important communication medium for individuals and organizations, but its open nature also makes it an attractive channel for unsolicited, deceptive, and malicious messages. Spam messages can impose substantial filtering and storage costs, whereas phishing messages may attempt to manipulate recipients into revealing credentials, financial information, or other sensitive data. Consequently, automated email classification has become an important application of machine learning and natural language processing (NLP).

Early research demonstrated that machine learning can be used to learn classification rules directly from email messages [1]. Classical text-processing methods subsequently established statistical representations such as term

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

weighting and TF-IDF as practical mechanisms for converting textual documents into numerical feature spaces [2]. Support Vector Machines (SVMs) have also been extensively investigated for text categorization because their formulation is well suited to high-dimensional sparse representations [3]. These approaches remain relevant when computational efficiency, reproducibility, and transparent feature construction are important considerations.

Phishing detection has received particular attention because malicious emails may be deliberately designed to resemble legitimate communication. Previous studies have investigated machine learning, NLP, deep learning, and ensemble approaches for detecting phishing messages [4]–[7]. For example, Alhogail and Alsabih investigated an NLP-based phishing detection approach using deep learning, while Bountakas and Xenakis examined ensemble learning with hybrid email features [5], [7]. Systematic reviews have also demonstrated the diversity of datasets, feature representations, and evaluation procedures used in phishing email detection [6]. These studies indicate that reported performance depends not only on the selected classifier but also on the characteristics of the dataset and the experimental protocol.

An important methodological issue in machine learning evaluation is data leakage. Leakage occurs when information that should not be available during model development influences the training or evaluation procedure [8]. Even when an explicit leakage mechanism is absent, inappropriate preprocessing, feature construction, or partitioning can allow information to flow between training and evaluation data. Such information sharing can produce overly optimistic estimates of predictive performance and reduce reproducibility [9].

Duplicate observations represent a particularly relevant source of overlap in large text collections. Identical or highly similar documents can occur because of repeated messages, templates, forwarded content, or aggregation from multiple source datasets. Research in NLP has demonstrated that duplicate and near-duplicate examples can influence evaluation and increase the overlap between training and evaluation collections [10]. More generally, validation strategies should account for dependence structures among observations rather than assuming that all individual records are statistically independent [11].

These issues are especially important for aggregated email datasets. If identical email texts are distributed between training and test sets, a classifier may partially exploit memorized lexical patterns rather than demonstrate generalization to previously unseen messages. Therefore, duplicate-aware partitioning provides a deterministic mechanism for preventing identical text groups from crossing evaluation boundaries. This does not imply that every duplicate necessarily represents leakage; rather, it provides a controlled procedure for ensuring that exact duplicate groups remain within a single partition.

The present study therefore focuses on the reliability of the evaluation protocol rather than on proposing a new classification algorithm. The study develops a leakage-aware multi-class classification framework that integrates duplicate auditing, removal of conflicting duplicate groups, group-aware partitioning, training-only feature construction, stratified group cross-validation, and a locked test set. A word-level TF-IDF representation with LinearSVC is used as the primary transparent classification pipeline, while an H2O AutoML branch provides a complementary automated-learning perspective [12]. The analysis is further supported by out-of-fold evaluation, locked-test evaluation, class-level error analysis, and bootstrap confidence intervals.

The main contributions of this study are as follows:

- A duplicate-aware preprocessing and partitioning procedure is developed for a large three-class email dataset.
- Conflicting duplicate groups are explicitly detected and removed before model development.
- A group-aware locked-test protocol is used to prevent exact duplicate groups from crossing the training-test boundary.
- Feature construction is restricted to the appropriate training data during cross-validation to reduce information leakage.
- A LinearSVC-based TF-IDF pipeline is evaluated using out-of-fold and locked-test predictions.

- A complementary H2O AutoML branch using a compressed SVD representation is evaluated under the same locked test set.
- Error analysis and bootstrap confidence intervals are used to characterize model behavior and uncertainty.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 describes the dataset, cleaning procedure, and duplicate audit. Section 4 presents the proposed leakage-aware framework. Section 5 describes the feature representation and classification pipelines. Section 6 explains the experimental setup and evaluation metrics. Section 7 presents the experimental results. Section 8 discusses their implications and limitations. Section 9 summarizes limitations and future work, and Section 10 concludes the study.

2. RELATED WORK

2.1. Machine Learning for Email Classification

Machine learning-based email classification has a long history. Cohen demonstrated the feasibility of learning rules for email classification, establishing an early foundation for automated email filtering [1]. Statistical term weighting subsequently became an established approach for representing textual documents numerically, with TF-IDF providing an effective representation for information retrieval and text classification [2].

SVM-based approaches are particularly relevant to text classification because textual representations are commonly sparse and high dimensional. Joachims showed that SVMs can effectively learn text classifiers with many relevant features [3]. This provides methodological motivation for using a linear SVM classifier with a high-dimensional word-level TF-IDF representation in the present study.

2.2. Phishing and Email Detection

Phishing detection has developed into an active research area within email security. Das et al. reviewed phishing research from a security perspective and emphasized challenges related to dataset quality, changing attacker behavior, and evaluation [4]. Alhogail and Alsabih investigated machine learning and NLP techniques for phishing email detection using textual email information [5]. Salloum et al. provided a systematic review of NLP-based phishing email detection and demonstrated the diversity of machine learning and NLP approaches applied to the problem [6].

Bountakas and Xenakis proposed HELPHED, a hybrid ensemble approach that combines different types of email features for phishing detection [7]. Such work demonstrates the potential benefits of more complex learning architectures and hybrid representations. However, it also illustrates the importance of considering the dataset and evaluation protocol when interpreting reported performance.

Among studies most closely aligned with the present three-class task, Doshi et al. [17] evaluated Ham, Phish, and Spam using an 8,218-message corpus assembled from Nazario and SpamAssassin and reported a maximum accuracy of 99.51% using their dual-layer architecture. However, their experimental protocol used a conventional 80/20 partition with class resampling, whereas the present study explicitly audits duplicate structure, enforces group-disjoint partitioning, fits text representations using training data only, and evaluates the final models on a locked test set. Therefore, the reported performance values are not directly comparable; the prior study is used here as task-level context rather than as a direct numerical benchmark.

2.3. Data Leakage and valuation Reliability

Data leakage can occur when information from outside the appropriate training partition influences model development [8]. Kaufman et al. described leakage as the introduction of information about the prediction target that should not legitimately be available during data mining [8]. Kapoor and Narayanan further discussed leakage as a contributor to reproducibility problems in machine-learning-based science and emphasized the importance of controlling information flow throughout the research pipeline [9].

For text classification, leakage can arise at multiple stages, including preprocessing, feature extraction, feature selection, model tuning, and evaluation. Consequently, leakage control should be considered as a property of the complete experimental pipeline rather than as a single preprocessing operation.

2.4. Duplicate Data and Group-Aware Evaluation

Duplicate and near-duplicate data have received increasing attention in NLP. Lee et al. demonstrated that large language-model datasets can contain substantial amounts of repeated and near-repeated text and that train-test overlap can affect evaluation [10]. Kambhatla et al. investigated train-evaluation overlap using nearest-neighbor methods and showed that overlap can vary considerably across datasets and tasks [13].

Cross-validation strategies also need to account for dependencies among observations. Roberts et al. demonstrated that ignoring structured dependencies can lead to overly optimistic estimates of predictive performance and discussed the use of non-random or blocked validation strategies when observations are not independent [11]. Although their work focuses on structured ecological data, the methodological principle is applicable more broadly: observations that share an underlying dependency should not automatically be treated as independent validation units.

Recent work specifically addressing email grouping further emphasizes that emails can share common sources, campaigns, or response structures [14]. Such grouping perspectives are relevant to spam and phishing because multiple messages may originate from the same underlying campaign while differing slightly in their textual content.

Exact duplicate detection provides a deterministic and reproducible first level of overlap control. However, near-duplicate detection addresses a broader class of overlap in which messages differ only by limited lexical, structural, or formatting modifications. The present study therefore explicitly distinguishes between exact duplicate control, which is implemented, and near-duplicate detection, which remains an important direction for future work.

2.5. Research Gap

The reviewed literature demonstrates extensive research on email classification, phishing detection, NLP representations, and machine learning algorithms. However, classification performance is often presented primarily as a property of the selected model, while the influence of duplicate structure and information flow throughout the evaluation pipeline receives comparatively less attention.

The present study addresses this methodological gap by treating evaluation reliability as an integral part of the classification framework. Rather than introducing a new classifier, it combines duplicate auditing, conflicting-duplicate removal, group-aware partitioning, training-only feature construction, stratified group cross-validation, locked-test evaluation, error analysis, and bootstrap uncertainty estimation.

3. DATASET AND PREPROCESSING

3.1. Dataset Description

The experiments use The Biggest Spam Ham Phish Email Dataset (250000+), a publicly available aggregated email dataset distributed through Kaggle [15]. The downloaded df.csv file contained 365,448 records with two principal fields: label and text. The three labels were encoded as 0 for Ham, 1 for Phish, and 2 for Spam.

The dataset is an aggregated collection constructed from multiple publicly available sources. Its large size makes it suitable for investigating duplicate structure and leakage-aware evaluation. At the same time, its aggregated provenance means that source-specific characteristics may influence the observed results. Therefore, the study treats the dataset as an experimental environment for evaluating the proposed methodology rather than as a universal representation of all operational email traffic.

The first step of the study was to preserve the original text content while removing records that could not be used meaningfully for classification. Only the label and text fields were retained. Missing textual values were converted to empty strings and stripped of leading and trailing whitespace. No stemming, punctuation removal, HTML normalization, or other semantic text transformations were performed during this initial cleaning stage. The main dataset statistics are summarized in Table I.

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

Statistic	Value
Raw records	365,448
Empty-text records removed	3
Records after empty-text removal	365,445
Unique text hashes	279,510
Duplicate groups	85,700
Duplicate rows beyond first occurrence	85,935
Largest duplicate group	6
Conflicting duplicate groups	14
Records in conflicting groups	42
Final records	365,403
Final unique text groups	279,496
Remaining conflicting groups	0

Table 1. Dataset and duplicate audit statistics.

As shown in Table I, the downloaded dataset contained substantially more duplicate structure than might be inferred from its nominal description. After removal of three empty records, 279,510 unique text hashes were identified among 365,445 records. A total of 85,700 duplicate groups were detected, with the largest group containing six identical records.

Importantly, 14 duplicate groups contained records assigned to more than one class. These groups involved 42 records and were removed entirely rather than assigning an arbitrary class label. This decision prevents a single textual instance from simultaneously representing conflicting class information and simplifies the interpretation of subsequent group-aware partitioning. After removing the conflicting groups, the final dataset contained 365,403 records and 279,496 unique text groups, with zero remaining conflicting duplicate groups.

3.2. Data Distribution

The final dataset contained 168,439 Ham messages, 42,838 Phish messages, and 154,126 Spam messages. Thus, the dataset is not perfectly balanced, with Phish representing the smallest class.

This imbalance was considered during classifier training through class weighting in the LinearSVC branch and class balancing in the H2O branch. Evaluation, however, was performed using both overall and class-sensitive metrics so that performance on the smaller Phish class was not hidden by aggregate accuracy.

3.3. Duplicate Detection

To identify exact duplicate groups, a deterministic SHA-256 hash was calculated from the normalized UTF-8 encoded text. For message (i), the duplicate-group identifier was defined as

$$g_i = \text{SHA256}(\text{text}_i)$$

where (g_i) represents the duplicate-group identifier and (text_i) denotes the normalized textual content.

Records with the same hash were treated as members of the same duplicate group. This procedure is deterministic and reproducible and avoids relying on a manually selected similarity threshold.

A duplicate group was considered conflicting if more than one class label occurred within the same text-hash group. Such groups were removed completely before data partitioning. This conservative treatment avoids assigning an artificial label to a text whose original records provide contradictory class information.

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

The procedure should be distinguished from near-duplicate detection. A single changed character, URL, identifier, or recipient name will generally result in a different SHA-256 value. Therefore, the current audit guarantees control over exact textual duplicates but does not claim to eliminate all semantically or structurally related messages.

3.4. Group-Aware Locked Partition

After cleaning and conflict removal, the final dataset was partitioned using five-fold StratifiedGroupKFold with stratification by class and grouping by duplicate identifier. A fixed random seed of 42 was used. Rather than generating an ordinary random 80/20 split, one of the five group-aware folds, Fold 0, was designated as the locked test set. This produced an approximately 80/20 partition while ensuring that duplicate groups did not cross the training-test boundary.

The resulting locked training set contained 292,365 messages, while the locked test set contained 73,038 messages. The training portion contained 223,597 unique text groups and the test portion contained 55,899 groups. The number of duplicate groups shared between the two partitions was zero. Table 2 summarizes the resulting partition.

Partition	Records	Ham	Phish	Spam	Unique groups
Locked training	292,365	134,880	34,212	123,273	223,597
Locked test	73,038	33,559	8,626	30,853	55,899
Total	365,403	168,439	42,838	154,126	279,496

Table 2. Locked train-test partition.

Table 2 shows that the class proportions remain broadly consistent between the two partitions. More importantly, the group counts confirm that the partition was performed at the duplicate-group level rather than by treating every row as an independent sampling unit.

The locked test set was not used for feature fitting, model selection, threshold selection, or parameter tuning. It was reserved for the final evaluation after model development.

4. LEAKAGE-AWARE CLASSIFICATION FRAMEWORK

4.1. Framework Overview

The proposed framework was designed around a simple principle: information available during model development must not cross into the final evaluation set. Figure 1 illustrates the complete experimental workflow.

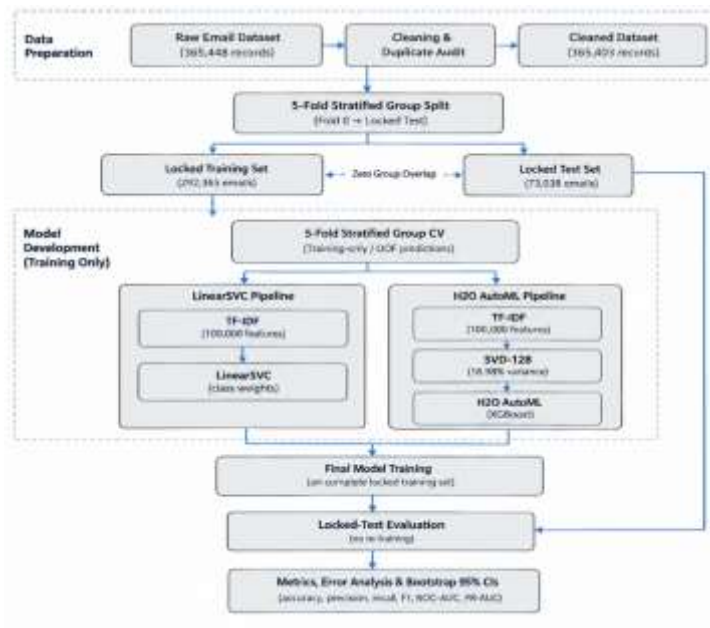


Figure 1. Overall leakage-aware framework for duplicate-aware multi-class email classification.

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

Figure 1 begins with the raw dataset and proceeds through cleaning, duplicate auditing, removal of conflicting groups, group-aware partitioning, cross-validation, feature construction, model training, and locked-test evaluation. The framework contains two model branches: the primary word-level TF-IDF and LinearSVC pipeline and the complementary SVD-based H2O AutoML pipeline.

The key structural property of the framework is that the locked test set is separated before final model development and remains untouched until the final evaluation. In addition, exact duplicate groups are prevented from crossing the training-test boundary and are also isolated during cross-validation.

4.2. Cross-Validation Protocol

The locked training portion was further evaluated using five-fold StratifiedGroupKFold cross-validation with shuffle enabled and random seed 42. The grouping variable remained the exact duplicate-group identifier. For each cross-validation fold, the TF-IDF vectorizer was fitted only on the training portion of that fold. The resulting representation was then applied to the held-out fold. This procedure prevents vocabulary statistics and inverse-document-frequency information from being estimated using validation examples.

The process generated out-of-fold (OOF) predictions for the complete locked training portion. These predictions were used to calculate cross-validation metrics without allowing any observation to be evaluated by a model trained on that observation. After completion of cross-validation, the final TF-IDF representation was fitted on the complete locked training set and the final LinearSVC model was trained on the same data. Only then was the locked test set transformed and evaluated.

4.3. Information-Flow Controls

The framework implements several explicit information-flow controls:

1. Conflicting duplicate groups are removed before partitioning.
2. Exact duplicate groups cannot cross the train-test boundary.
3. Exact duplicate groups cannot cross cross-validation folds.
4. TF-IDF statistics are estimated only from the corresponding training data during cross-validation.
5. OOF predictions are generated from models that did not train on the corresponding held-out observations.
6. The locked test set is excluded from model development and evaluated once after the final model is established.

These controls collectively define the term **leakage-aware evaluation** used throughout this paper. Duplicate-aware partitioning is therefore treated as one component of the broader leakage-control strategy rather than as a synonym for leakage itself.

5. FEATURE REPRESENTATION AND CLASSIFIERS

5.1. Word-Level TF-IDF

The primary feature representation was based on word-level Term Frequency-Inverse Document Frequency (TF-IDF), a widely used representation for text classification [2]. The vectorizer used unigram and bigram features with the following configuration:

- analyzer: word;
- n-gram range: (1, 2);
- minimum document frequency: 3;
- maximum document frequency: 0.98;

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

- maximum number of features: 100,000;
- sublinear term-frequency scaling;
- Unicode accent stripping;
- lowercase conversion;
- 32-bit floating-point representation.

The 100,000-feature limit was used to provide a high-dimensional but computationally manageable representation. Importantly, the vectorizer was fitted inside each cross-validation training fold rather than before the fold split.

5.2. LinearSVC Classifier

The primary classifier was LinearSVC, selected because linear support vector methods are well established for high-dimensional text classification [3]. The classifier used:

- ($C=1.0$);
- balanced class weighting;
- maximum iterations = 2,000;
- random seed = 42.

Balanced class weighting was used to reduce the influence of the class imbalance in the training data. The LinearSVC produces decision scores rather than calibrated class probabilities. Therefore, ROC-AUC and average precision were calculated using one-vs-rest decision scores for each class and then macro-averaged.

5.3. H2O AutoML Branch

A second branch was implemented using H2O AutoML to provide a complementary automated-learning comparison [12]. Because the original word-level TF-IDF representation can contain up to 100,000 dimensions, Truncated Singular Value Decomposition (SVD) was applied before H2O training.

The SVD configuration used 128 components, the randomized algorithm, five iterations, and random seed 42. The resulting representation contained $292,365 \times 128$ training observations and $73,038 \times 128$ test observations. The 128 components retained approximately 16.98% of the total variance.

The relatively low retained-variance value should be interpreted in the context of the purpose of dimensionality reduction. The SVD representation was not selected with the objective of maximizing variance retention. Instead, it was used to substantially reduce the dimensionality of the original sparse TF-IDF representation and provide a computationally practical input for the automated-learning branch under the predefined runtime and resource constraints. The retained variance is therefore a characteristic of the selected compressed representation rather than an independent measure of the quality of the H2O branch.

The H2O AutoML configuration used a maximum runtime of 1,800 s, a maximum model budget of 15 models, random seed 42, class balancing, and AUC-based model ranking. Deep Learning was excluded. Only one model completed within the allocated runtime, namely an XGBoost model. A GLM model did not complete before the runtime limit. Consequently, the H2O experiment represents a limited comparative automated-learning branch rather than an exhaustive benchmark of the H2O model space.

5.4. Scope of Transformer-Based Models

Transformer-based language models have demonstrated strong capabilities in a broad range of NLP tasks. BERT, for example, introduced bidirectional Transformer-based language representations that can be fine-tuned for downstream tasks [16].

Such models were not included in the present experimental scope because the primary objective was to investigate the reliability of the evaluation protocol rather than to conduct an exhaustive comparison of modern language architectures. The selected LinearSVC pipeline also provides a transparent and computationally efficient baseline for evaluating a dataset containing more than 365,000 records.

This scope limitation does not imply that Transformer-based approaches are unsuitable for the task. Instead, evaluating models such as BERT under the same duplicate-aware and leakage-controlled protocol represents a direct and important extension of the present framework. Such an experiment would allow the effect of model architecture to be studied while maintaining consistent controls over duplicate overlap, feature-learning information flow, and locked-test evaluation.

6. EXPERIMENTAL SETUP AND EVALUATION

6.1. Experimental Environment

The experiments were implemented using Python-based machine learning tools. The primary text classification branch used scikit-learn components for TF-IDF vectorization, group-aware cross-validation, LinearSVC classification, and evaluation. The automated-learning branch used H2O AutoML with a compact SVD representation. The experimental protocol used fixed random seed 42 wherever supported. Model artifacts, predictions, evaluation outputs, and configuration information were stored to support reproducibility.

6.2. Evaluation Metrics

The main metrics were accuracy, macro precision, macro recall, macro F1, weighted F1, Phish precision, Phish Performance was evaluated using accuracy, macro precision, macro recall, macro F1-score, weighted F1-score, class-specific precision, recall, and F1-score.

Accuracy is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

for binary classification and is generalized to the multi-class setting as the proportion of correctly classified observations. For each class, precision and recall are defined as

$$Precision = \frac{TP}{TP + FP}$$

and

$$Recall = \frac{TP}{TP + FN}$$

The F1-score is defined as

$$F1 = 2 \frac{Precision \times Recall}{Precision + Recall}$$

Macro averaging gives each class equal weight, whereas weighted averaging weights each class according to its number of observations. Both are reported because the dataset is not perfectly balanced. For ranking-based evaluation, one-vs-rest ROC-AUC and average precision were calculated separately for each class and macro-averaged.

6.3. Out-of-Fold Evaluation

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

The primary LinearSVC model was evaluated on the locked training portion using five-fold StratifiedGroupKFold cross-validation. The resulting OOF predictions provide an estimate of model behavior on observations not used to train the corresponding fold-specific model.

This evaluation is distinct from the locked-test evaluation. OOF predictions are used for development-stage assessment, whereas the locked test is reserved for the final reported generalization estimate under the predefined partition.

6.3. Bootstrap Confidence Intervals

To characterize uncertainty in the locked-test estimates, nonparametric bootstrap confidence intervals were calculated using 2,000 bootstrap resamples with random seed 42. Each bootstrap sample was drawn from the locked test set with replacement, and the relevant metric was recalculated. The 2.5th and 97.5th percentiles of the resulting distributions were used to form the 95% confidence intervals. Bootstrap intervals were calculated for accuracy, macro precision, macro recall, macro F1, weighted F1, and the principal Phish-class metrics.

7. RESULTS AND ANALYSIS

7.1. Out-of-Fold Results

The OOF results of the primary LinearSVC pipeline are presented in Table 3. These results were obtained using five-fold StratifiedGroupKFold cross-validation on the locked training portion.

Metric	OOF Result
Accuracy	97.92%
Macro Precision	97.11%
Macro Recall	97.35%
Macro F1	97.23%
Weighted F1	97.92%
Phish Precision	94.70%
Phish Recall	95.65%
Phish F1	95.18%
Macro ROC-AUC	99.81%
Macro PR-AUC	99.52%

Table 3. Out-of-fold performance of the LinearSVC pipeline.

Table III shows consistently high OOF performance. The macro F1-score of 97.23% indicates that the classifier maintained strong performance across the three classes rather than relying exclusively on the larger Ham and Spam classes. The Phish class achieved an F1-score of 95.18%, which is lower than the aggregate metrics but remains high relative to the overall classification performance.

The close relationship between the OOF results and the final locked-test results also provides a useful indication that the observed performance is not substantially altered when the final model is evaluated on the previously untouched test partition.

7.2. Locked-Test Performance

After cross-validation was completed, the final LinearSVC model was trained on the complete locked training set. The locked test set was then evaluated once.

The H2O branch was evaluated on the same locked test partition. The resulting comparison is presented in Table 4.

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

Metric	LinearSVC + TF-IDF	H2O AutoML + SVD-128
Accuracy	98.00%	97.47%
Macro Precision	97.22%	97.00%
Macro Recall	97.42%	96.98%
Macro F1	97.32%	96.99%
Weighted F1	98.00%	97.47%
Phish Precision	94.86%	95.62%
Phish Recall	95.70%	95.48%
Phish F1	95.28%	95.55%
Macro ROC-AUC	99.81%	99.84%
Macro PR-AUC	99.53%	99.61%

Table 4. Locked-test performance comparison

Table IV shows that the two pipelines achieved similar overall performance but exhibited different metric profiles. The LinearSVC pipeline achieved higher accuracy, macro precision, macro recall, macro F1, and weighted F1. The H2O branch achieved slightly higher Phish precision, Phish F1, macro ROC-AUC, and macro PR-AUC. These results should be interpreted at the pipeline level rather than as a classifier-only comparison. LinearSVC operates directly on the 100,000-dimensional word-level TF-IDF representation, whereas the H2O branch operates on a 128-dimensional SVD representation. Therefore, the observed differences cannot be attributed exclusively to the learning algorithms. The lower aggregate performance of the H2O branch may be influenced by information compression in the SVD representation, but the current experiments do not isolate the effect of SVD dimensionality experimentally. Consequently, the results do not establish the absolute superiority of one classifier family over another.

7.3. Confusion Matrix Analysis

To examine class-specific behavior beyond aggregate metrics, the locked-test confusion matrices for both pipelines are presented in Figure 2.

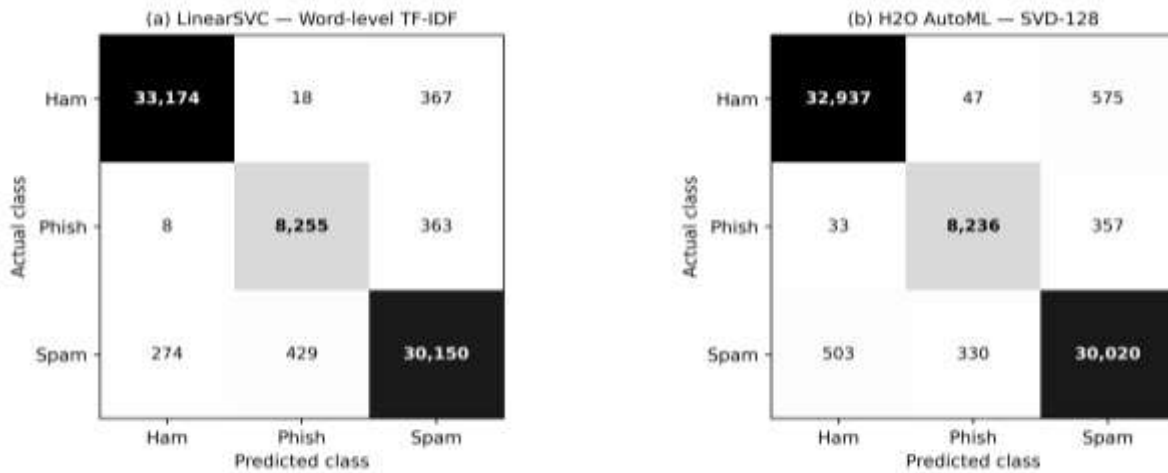


Figure 2. Locked-test confusion matrices: (a) LinearSVC with word-level TF-IDF and (b) H2O AutoML with SVD-128.

Figure 2(a) shows that the LinearSVC pipeline correctly classified 33,174 Ham messages, 8,255 Phish messages, and 30,150 Spam messages. The largest error category was Spam classified as Phish, with 429 messages, followed by Ham classified as Spam with 367 messages and Phish classified as Spam with 363 messages. Figure 2(b) shows that the H2O branch correctly classified 32,937 Ham messages, 8,236 Phish messages, and 30,020 Spam messages. Its largest error category was Ham classified as Spam, with 575 messages, followed by Spam classified as Ham, with 503 messages. The confusion matrices indicate that the majority of errors involve interactions between Spam

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

and the other two categories. This pattern is consistent with the lexical overlap that can occur between unsolicited commercial messages, deceptive messages, and legitimate messages containing promotional or transactional language.

7.4. Error Analysis

The detailed error distribution of the LinearSVC pipeline is shown in Table 5.

Error Pair	Count	% of Errors	% of True Class
Spam → Phish	429	29.40%	1.39%
Ham → Spam	367	25.15%	1.09%
Phish → Spam	363	24.88%	4.21%
Spam → Ham	274	18.78%	0.89%
Ham → Phish	18	1.23%	0.05%
Phish → Ham	8	0.55%	0.09%
Total	1,459	100.00%	—

Table 5. LinearSVC misclassification pairs on the locked test set

As shown in Table 5, the LinearSVC classifier produced 1,459 misclassified messages, corresponding to 1.9976% of the 73,038-message locked test set. Spam-to-Phish errors were the largest category and represented 29.40% of all classification errors. Ham-to-Spam and Phish-to-Spam errors accounted for 25.15% and 24.88%, respectively. The percentage of true-class errors provides additional context. Phish had the highest class-specific error rate, with 4.21% of true Phish messages being classified as Spam and smaller proportions assigned to Ham. This is consistent with the observation that Phish is the smallest class and may share lexical characteristics with Spam. The error distribution is visualized in Figure 3.

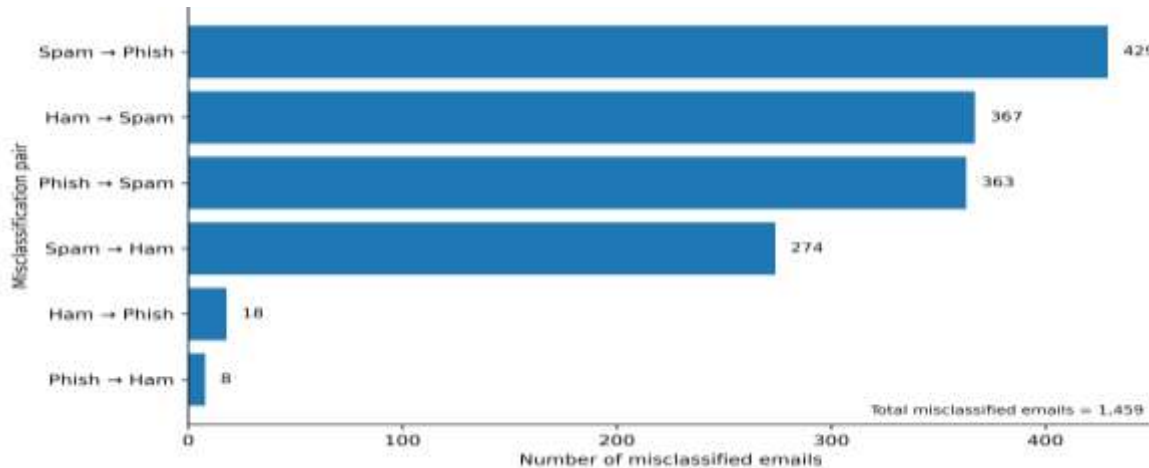


Figure 3. Distribution of LinearSVC misclassification pairs on the locked test set.

Figure 3 confirms that the dominant errors are concentrated in four class transitions: Spam→Phish, Ham→Spam, Phish→Spam, and Spam→Ham. The relatively small number of Ham→Phish and Phish→Ham errors suggests that direct confusion between Ham and Phish is substantially less frequent than confusion involving Spam.

These patterns provide useful information beyond a single aggregate accuracy value. In particular, they indicate that future improvements may benefit from representations capable of distinguishing malicious intent from lexical characteristics shared by promotional, transactional, and unsolicited messages.

7.5. Statistical Uncertainty

The bootstrap confidence intervals for the principal locked-test metrics are presented in Table 6.

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

Model	Metric	Estimate	95% CI
LinearSVC	Accuracy	98.00%	[97.90, 98.10]
	Macro Precision	97.22%	[97.06, 97.39]
	Macro Recall	97.42%	[97.27, 97.58]
	Macro F1	97.32%	[97.18, 97.47]
	Weighted F1	98.00%	[97.90, 98.10]
	Phish Precision	94.86%	[94.39, 95.33]
	Phish Recall	95.70%	[95.25, 96.13]
	Phish F1	95.28%	[94.95, 95.61]
H2O-SVD128	Accuracy	97.47%	[97.36, 97.58]
	Macro Precision	97.01%	[96.84, 97.16]
	Macro Recall	96.98%	[96.81, 97.14]
	Macro F1	96.99%	[96.84, 97.13]
	Weighted F1	97.47%	[97.36, 97.58]
	Phish Precision	95.63%	[95.21, 96.04]
	Phish Recall	95.48%	[95.04, 95.92]
	Phish F1	95.55%	[95.24, 95.86]

Table 6. Bootstrap 95% confidence intervals for locked-test performance.

Table 6 indicates relatively narrow confidence intervals for the reported metrics, which is expected given the large locked-test set. For LinearSVC, the 95% bootstrap interval for accuracy ranges from 97.90% to 98.10%, while the corresponding interval for macro F1 ranges from 97.18% to 97.47%.

The H2O branch exhibits similarly narrow intervals. These intervals quantify sampling uncertainty within the evaluated locked-test set; they should not be interpreted as evidence that the reported performance necessarily generalizes to independent datasets with different distributions.

8. DISCUSSION

8.1. Reliability of the Evaluation Protocol

The central methodological result of the study is the explicit separation between model development and final evaluation. The dataset audit demonstrated that duplicate structure was substantial: 85,700 duplicate groups were identified, and 14 groups contained conflicting class labels.

If records had been partitioned independently without grouping, identical text could potentially appear in different partitions. The present procedure prevents this by defining duplicate groups before partitioning and using those groups as the grouping variable during both train-test partitioning and cross-validation.

The resulting zero group overlap between the locked training and test sets provides a clear and reproducible property of the evaluation protocol. It does not establish that all forms of information leakage have been eliminated, but it demonstrates that exact duplicate-group overlap was explicitly controlled.

8.2. Information Flow and Feature Construction

Another important property of the framework is that TF-IDF fitting occurs within the appropriate training partition during cross-validation. This prevents validation documents from contributing to the vocabulary statistics used to construct their own evaluation representation. After cross-validation, the final vectorizer was fitted on the complete locked training set. The locked test set remained unused until the final evaluation.

This distinction is important because leakage control is not achieved solely by separating the raw records. A pipeline can still leak information if feature construction or preprocessing is performed globally before cross-validation.

8.3. Dataset Provenance and External Validity

The evaluated dataset is a publicly available aggregated collection obtained from Kaggle and combines email records originating from multiple sources [15]. This provides a large experimental resource but also limits the extent to which the results can be interpreted as representative of independent operational email environments.

Source-specific collection procedures, class definitions, duplication patterns, and distributional characteristics may influence classification performance. Accordingly, the primary objective of the present study is methodological: to develop and evaluate a leakage-aware classification and evaluation framework on a large dataset in which duplicate structure can be explicitly audited and controlled. The reported performance should therefore be interpreted as performance on the evaluated dataset rather than as a universal estimate of email-classification performance. Independent evaluation on established public datasets such as Enron and other spam and phishing collections would provide a stronger assessment of external validity. Applying the same duplicate-aware and leakage-controlled protocol across independent sources would help determine whether the observed evaluation properties remain consistent under different data distributions.

8.4. Interpretation of the LinearSVC Results

The strong LinearSVC performance demonstrates that a conventional high-dimensional TF-IDF representation can provide an effective baseline for multi-class email classification. The model achieved 98.00% accuracy and 97.32% macro F1 on the locked test set.

However, the contribution of the study should not be interpreted simply as achieving a high classification score. The main methodological value is that this performance was obtained after duplicate auditing, group-aware partitioning, training-only feature construction, group-aware cross-validation, and locked-test evaluation. This distinction is important because a high score obtained from a less controlled experimental protocol may not provide the same level of evidence regarding generalization.

8.6. Interpretation of the H2O Branch and SVD Limitation

The error analysis provides further insight into the classification problem. The largest LinearSVC error category was Spam→Phish, followed by Ham→Spam and Phish→Spam.

Such errors may reflect overlap in the lexical and structural characteristics of email classes. Spam and phishing messages can both contain urgency, promotional language, requests for action, links, and other persuasive elements. Legitimate transactional or promotional messages can also contain terms commonly associated with Spam. Because the present study uses textual content as its principal input, it does not explicitly model sender identity, domain reputation, URL structure, campaign membership, temporal information, or other metadata. Incorporating such information may provide additional discriminative signals in future work.

8.7. Practical Implications

The results suggest that evaluation design should be considered alongside classifier selection when developing large-scale email classification systems. A conventional classifier can achieve strong performance when the representation is appropriate, but the credibility of that performance depends on whether the evaluation protocol prevents unintended overlap and information sharing. The proposed framework is therefore applicable beyond the specific LinearSVC and H2O configurations examined here. The same principles can be applied to other classifiers, neural networks, and Transformer-based models.

8.8. Recommendations

Based on the evaluated workflow, three methodological recommendations can be made. First, duplicate auditing should be performed before train-test partitioning when working with aggregated text datasets. Second, text representations and other preprocessing transformations should be fitted using training data only, particularly during cross-validation. Third, final evaluation should rely on a locked test set and should report class-level metrics and uncertainty rather than accuracy alone. When source, sender, campaign, or temporal metadata are available, these attributes should also be considered as potential grouping variables in future evaluations

9. LIMITATIONS AND FUTURE WORK

Several limitations should be considered when interpreting the results. The primary limitation stems from evaluating the methodology on a single publicly available aggregated dataset; although the dataset is large, external validity cannot be established from one collection alone. Consequently, future work should apply the same evaluation protocol to independent datasets with different collection procedures and class distributions.

Regarding duplicate handling, the duplicate audit focused exclusively on exact textual duplicates identified via SHA-256 hashing. Near-duplicate messages—such as repeated email templates with minor lexical, formatting, recipient, identifier, or URL modifications—may remain in the dataset because any subtle textual variation yields a distinct SHA-256 hash. Future work should therefore extend the duplicate audit using scalable techniques such as MinHash and Locality-Sensitive Hashing (LSH), alongside similarity-based or semantic approaches, to quantify the prevalence and impact of partially overlapping messages. Furthermore, the group definition in this study relies solely on exact textual identity, as other dependency structures (such as common sender, domain, campaign, thread, or temporal origin) were unavailable in the evaluated representation; future datasets incorporating such metadata could facilitate broader group-aware evaluation protocols.

On the modeling front, the H2O branch employed a 128-component SVD representation retaining approximately 16.98% of the variance. Because the principal objective of SVD was computational compression rather than variance maximization, the chosen dimensionality may influence classification performance—an effect not isolated in the current experiments, though future studies could evaluate multiple SVD dimensions under fixed computational constraints. Additionally, the H2O AutoML execution was bounded by a 1,800-second runtime limit and a 15-model budget, resulting in only one model completing within the allocated timeframe; thus, the reported H2O performance should not be construed as an exhaustive benchmark of the H2O model space. Consequently, the comparison between LinearSVC and H2O represents a pipeline-level evaluation rather than a strict classifier-to-classifier benchmark, as LinearSVC operates on the full 100,000-dimensional TF-IDF space whereas H2O utilizes the compressed 128-dimensional space. Future work should perform more controlled comparisons by holding both representation and computational budget constant.

Finally, Transformer-based architectures were outside the scope of this study. Evaluating BERT and related Transformer models under this duplicate-aware and leakage-controlled protocol remains a crucial future direction, enabling architectural comparisons under rigorous information-flow controls. Ultimately, while the proposed evaluation framework establishes robust internal methodological control, it does not guarantee universal real-world generalization, and performance metrics should be interpreted strictly within the context of the evaluated dataset and experimental protocol.

10. CONCLUSION

This study presented a leakage-aware framework for multi-class email classification, establishing rigorous, explicit controls over duplicate structure and information flow throughout the experimental pipeline. The proposed methodology systematically integrates duplicate auditing, the removal of conflicting duplicate groups, group-aware partitioning, training-isolated feature construction, stratified group cross-validation, locked-test evaluation, error analysis, and bootstrap confidence intervals. Applied to the evaluated dataset, the initial audit uncovered substantial exact-duplicate patterns, identifying 85,700 duplicate groups alongside 14 conflicting groups containing 42 records. Following the exclusion of empty and conflicting entries, a cleaned dataset of 365,403 messages was partitioned into an approximate 80/20 group-aware split, yielding a locked test set of 73,038 messages with verified zero duplicate-group overlap between the training and testing partitions.

On this held-out test set, the primary word-level TF-IDF and LinearSVC pipeline achieved 98.00% accuracy and a 97.32% macro F1 score. Concurrently, the complementary H2O AutoML pipeline, operating on a 128-component SVD representation, achieved 97.47% accuracy and a 96.99% macro F1 score while obtaining slightly higher macro ROC-AUC and PR-AUC metrics. Owing to fundamental differences in feature representations and learning configurations, these comparative findings were interpreted strictly at the pipeline level rather than as a classifier-only benchmark.

Leakage-Aware Multi-Class Email Classification Using Duplicate-Aware Evaluation and Automated Machine Learning

Ultimately, the principal contribution of this work is methodological rather than the introduction of a new classification algorithm. By demonstrating how duplicate-aware partitioning and strict information-flow controls can be seamlessly unified within a fully reproducible multi-class email classification workflow, this study provides a standardized benchmarking foundation for conventional, automated, and emerging Transformer-based classifiers under a unified leakage-controlled protocol. To build upon these findings and establish broader external validity, future research should extend the framework toward near-duplicate detection, evaluation across independent external datasets, integration of richer campaign- or domain-level metadata, controlled feature-dimensionality experiments, and the benchmarking of Transformer architectures under identical experimental controls.

References

- [1] Cohen, W. W. (1996). Learning rules that classify e-mail. In AAAI Spring Symposium on Machine Learning and Information Access.
- [2] Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, 24(5), 513–523. [https://doi.org/10.1016/0306-4573\(88\)90021-0](https://doi.org/10.1016/0306-4573(88)90021-0)
- [3] Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning (ECML) (Lecture Notes in Computer Science, Vol. 1398, pp. 137–142)*. Springer. <https://doi.org/10.1007/BFb0026683>
- [4] Das, A., Baki, S., El Aassal, A., Verma, R., & Dunbar, A. (2020). SoK: A comprehensive reexamination of phishing research from the security perspective. *IEEE Communications Surveys & Tutorials*, 22(1), 671–708. <https://doi.org/10.1109/COMST.2019.2957750>
- [5] Alhogail, A., & Alsabih, A. (2021). Applying machine learning and natural language processing to detect phishing email. *Computers & Security*, 110, Article 102414. <https://doi.org/10.1016/j.cose.2021.102414>
- [6] Salloum, S. A., Gaber, T., Vadera, S., & Shaalan, K. (2022). A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access*, 10, 65703–65727. <https://doi.org/10.1109/ACCESS.2022.3183083>
- [7] Bountakas, P., & Xenakis, C. (2023). HELPHED: Hybrid Ensemble Learning PHishing Email Detection. *Journal of Network and Computer Applications*, 210, Article 103545. <https://doi.org/10.1016/j.jnca.2022.103545>
- [8] Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data*, 6(4), Article 15. <https://doi.org/10.1145/2382577.2382579>
- [9] Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, 4(9), Article 100804. <https://doi.org/10.1016/j.patter.2023.100804>
- [10] Lee, K., Ippolito, D., Nystrom, A., Zhang, C., Eck, D., Callison-Burch, C., & Carlini, N. (2022). Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (pp. 8424–8445)*. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2022.acl-long.577>
- [11] Roberts, D. R., Bahn, V., Ciuti, S., Boyce, M. S., Elith, J., Guillera-Aroita, G., et al. (2017). Cross-validation strategies for data with temporal, spatial, hierarchical, or phylogenetic structure. *Ecography*, 40(8), 913–929. <https://doi.org/10.1111/ecog.02881>
- [12] LeDell, E., & Poirier, S. (2020). H2O AutoML: Scalable automatic machine learning. In *7th ICML Workshop on Automated Machine Learning*.
- [13] Kambhatla, G., Nguyen, T., & Choi, E. (2023). Quantifying train-evaluation overlap with nearest neighbors. In *Findings of the Association for Computational Linguistics: ACL 2023 (pp. 2905–2920)*. Association for Computational Linguistics. <https://doi.org/10.18653/v1/2023.findings-acl.183>

- [14] Saka, T., Vanica, K., & Kökcüyan, N. (2025). SoK: Grouping spam and phishing email threats for smarter security. *IEEE Access*, 13, 54938–54959. <https://doi.org/10.1109/ACCESS.2025.3555157>
- [15] Sharma, A. (2026). The biggest spam ham phish email dataset (250000+) (Version 1) [Data set]. Kaggle.
- [16] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* (Vol. 1, pp. 4171–4186). Association for Computational Linguistics. <https://doi.org/10.18653/v1/N19-1423>
- [17] J. Doshi, K. Parmar, R. Sanghavi, and N. Shekokar, “A comprehensive dual-layer architecture for phishing and spam email detection,” *Computers & Security*, vol. 133, Art. no. 103378, 2023, doi: 10.1016/j.cose.2023.103378.