

Article

Classical-First Selective Cascades for Resource-Constrained Phishing Email Detection

Abdulaziz Alajaji 

Department of Information Systems, College of Computer and Information Sciences, King Saud University, Riyadh 11451, Saudi Arabia; aalajaji@ksu.edu.sa

Abstract

Phishing email detection is a practical requirement for human-operated security and administration workflows, including mail gateways, security operations centre (SOC) triage queues, cloud dashboards, firmware-management portals, and other operational interfaces. In Internet-of-Things (IoT) and wireless sensor network (WSN) operations, compromise of such human-operated interfaces can open an initial-access path into the sensor-network management plane; however, the present study evaluates email body classifiers on public phishing email corpora rather than on-device IoT/WSN hardware. Email filtering must often run on low-cost servers, where running and storing a transformer for every message is costly in compute and memory, and opaque scores are hard to audit. We compare four model families on four public phishing email corpora: hand-engineered classifiers, high-vocabulary classical models, frozen-transformer classifiers, and a CPU-feasible fine-tuned DistilBERT reference. The strongest low-cost model is a Platt-calibrated Linear SVM with high-vocabulary TF-IDF features. It reaches $F1 = 0.961$ on the primary corpus, slightly exceeds the frozen DistilBERT baseline and is statistically indistinguishable from the fine-tuned DistilBERT reference in-domain, and requires 0.23 MB on disk with about 8 ms per email. This shows that transformer inference is not required for competitive accuracy on the evaluated in-domain corpora. We then evaluate the Calibrated Selective Cascade (CSC) as an operational routing layer: a calibrated low-cost arm handles most messages, while a validation-selected uncertainty band is deferred to a transformer arm. With the strong SVM arm, CSC yields only marginal in-domain $F1$ gains, but provides a tunable latency/deferral trade-off parameter and a monitorable drift signal. On an exploratory synthetic stress set of stylistically neutral, LLM-style phishing, no evaluated body-only model exceeds $F1$ around 0.46; closing this gap would require header-, URL-, identity-, and attachment-level signals.

Keywords: phishing email detection; email security; selective classification; cascade architecture; calibration; classical machine learning; transformer inference; resource-constrained inference; operational security; interpretable machine learning



Academic Editor: Domenico Ursino

Received: 12 June 2026

Revised: 8 July 2026

Accepted: 9 July 2026

Published: 11 July 2026

Copyright: © 2026 by the author.

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.

1. Introduction

Phishing has remained the leading initial-access vector in industry threat reports for several consecutive years [1,2]. The email modality alone accounts for the majority of reported credential-theft incidents, and the cost per successful phishing-driven breach continues to rise in step with the broader cybercrime economy [1]. Defensive measures are layered: gateway-level filtering removes the bulk of obviously malicious messages; in-mailbox classifiers tag suspicious mail and surface warning banners to end users; and

security-operation workflows aggregate the residual signal for triage. Each layer has different latency, interpretability, and false-positive budgets. The classifiers we study here are intended for *either* of the first two layers, both of which prefer fast, low-cost, and explainable models that can be retrained quickly when attackers pivot.

In Internet-of-Things (IoT) and wireless sensor network (WSN) deployments, the attack surface extends beyond sensor nodes and wireless protocols to the administrators, gateway operators, cloud dashboards, device-management consoles, firmware-update portals, and incident-response mailboxes that operate them. In industrial IoT control systems, a phishing email that compromises an operator account can reach the process-monitoring and control commands that govern physical equipment such as wirelessly connected furnaces and production lines [3], threatening system stability. In smart-grid operations, stolen operator credentials can expose the energy-management and metering platforms that coordinate generation and load [4], allowing an attacker to manipulate monitoring data or disrupt control functions. Comparable risks arise in intelligent transportation and healthcare IoT management portals, where unauthorised access to the management plane has direct operational consequences.

More generally, a successful phishing email can expose credentials for sensor gateways, MQTT brokers, cloud IoT platforms, and monitoring dashboards [5,6], opening an initial-access path into the management plane that bypasses the cryptographic and routing protections of the radio and network layers [7]. Email security filtering is therefore a relevant complementary layer in IoT/WSN security architectures, especially when the classifier must run on low-cost edge servers or gateway-adjacent infrastructure [8].

The academic literature on phishing email detection has bifurcated over the past five years. At one end, transformer encoders such as BERT, DistilBERT and RoBERTa are fine-tuned on labelled phishing corpora and achieve F1 scores in the 0.92–0.98 range [9–11]. At the other end, classical pipelines built on TF-IDF or n-grams remain the standard baseline. The former are accurate but expensive: training requires GPU acceleration, gateway-throughput inference requires either a GPU farm or aggressive batching, and any single prediction is in practice opaque without an additional explanation layer such as SHAP or LIME [12]. The latter are auditable, but as we will demonstrate, their reported performance depends strongly on choices that the literature has not systematically controlled, in particular, the interaction between *imbalance handling* and *feature engineering*.

The deployment question is therefore selective. It includes not only which classifier has the highest F1, but which messages require the expensive model and which can be handled by a calibrated, inspectable low-cost model at an acceptable cost in detection quality. The Calibrated Selective Cascade (CSC) proposed in this paper addresses this routing question directly.

This paper targets the human-facing email layer around IoT/WSN operations, not packet-level or sensor-stream security.

1.1. Operational Security Context and Scope

This paper evaluates the model capacity required for resource-constrained phishing email detection in the operational layer around IoT/WSN deployments, such as gateway-adjacent servers, security operations centre (SOC) mailboxes, administrator inboxes, and cloud-management workflows. The central finding is empirical. On four public corpora, a Platt-calibrated high-vocabulary classical model slightly exceeds the frozen DistilBERT baseline and is statistically indistinguishable from the fine-tuned reference in F1 at a fraction of their latency and disk cost. The same model also transfers better than hand-engineered features, resists low-cost stylistic evasions, and remains well calibrated. On top of this model, we build the Calibrated Selective Cascade (CSC), which keeps the calibrated

classical model as a low-cost sparse-linear arm and defers only a validation-selected band of uncertain messages to a transformer arm. CSC adapts calibrated selective routing to phishing email detection under low-cost operational constraints. Its primary contribution is deployment control: calibrated deferral, latency management, and drift monitoring under resource constraints. This paper makes the following contributions.

1. *Comparative evaluation of classical and transformer-based phishing classifiers.* A four-corpus benchmark of 90,786 emails after cross-corpus deduplication (Section 3.1). We show that a calibrated high-vocabulary classical model slightly exceeds frozen DistilBERT and is statistically tied with the fine-tuned DistilBERT reference in F1 (0.961 on Dataset A), at 0.23 MB and about 8 ms per email. A deployment stress comparison covers cross-corpus transfer, attacker-side robustness, and calibration (Sections 4.2.3 and 4.5).
2. *Classical-first Calibrated Selective Cascade.* A calibrated probability-band deferral policy with the strong classical model as the low-cost sparse-linear arm and a transformer arm, evaluated under a strict validation-only thresholding protocol at five operating points (Sections 3.4 and 4.5.3). We quantify its F1, latency, deferral rate, and novelty-monitoring behaviour.
3. *Cross-corpus transfer.* A 4×4 train $\times$ test cross-corpus transfer matrix for both the hand-engineered and the strong classical model, plus a union-trained reference and a per-source decomposition of the seven-source Dataset C that exposes within-source heterogeneity concealed by single-number benchmarks (Sections 4.4.2, 4.4.3 and 4.5.1).
4. *Robustness, calibration, and LLM-style phishing.* Attacker-side stylistic stress tests, a calibration analysis, and an exploratory LLM-style stress set (Sections 4.6.1, 4.7.1 and 4.7.2). On the LLM-style set, F1 falls to about 0.46 for every evaluated model, both classical and transformer.
5. *Operational-context case study.* An evaluation on operator-targeted phishing scenarios and a real operations-themed email subset that situates the method in human-facing operational security workflows (Section 4.8).

A supporting empirical finding that justifies the low-cost arm is that *feature representation has a substantially larger effect on phishing-class F1 than imbalance-handling strategy* (Section 4.1.3). The reproducibility package, including feature matrices, model outputs, cascade sweeps, statistical tests, and figure scripts, is available from the corresponding author upon reasonable request.

1.2. Scope

The evaluated scope is English email body classification for phishing vs. legitimate email in operational security contexts; the protected assets are the human-operated interfaces surrounding the sensor-network deployments (administrator inboxes, gateway-management workflows, firmware-update portals, cloud dashboards, SOC triage mailboxes). URL-only, image-based, header-only, vishing, and smishing detection, like the packet-level WSN tasks excluded above, remain separate tasks requiring different feature pipelines, datasets, and operating thresholds. The four datasets are all publicly available on the Hugging Face Hub; after the deduplication and minimum-length filters described in Section 3.1, we work with 90,786 emails across the four corpora.

1.3. Organisation

Section 2 reviews related work. Section 3 describes the datasets and preprocessing, the low-cost and transformer arms, the Calibrated Selective Cascade, and the evaluation protocol and statistical analysis. Section 4 presents the results: low-cost arm selection, transformer baselines, the cascade, cross-corpus generalisation, calibration and operational

thresholds, and robustness and error analysis. Section 5 discusses deployment implications, limitations, and ethical considerations; Section 6 concludes.

2. Related Work

2.1. Phishing Email Corpora

Public phishing email corpora fall into three groups. *Static phishing archives*: Nazario's "phishingcorpus" [13] is the longest-running. *Legitimate mail archives*: the Enron corpus [14] and the SpamAssassin public corpus [15] are the standard sources of legitimate ham. *Composite corpora*: the IWSPA-AP 2018 shared-task release [16] and several Kaggle and Hugging Face redistributions mix the two. The zefang-liu/phishing-email-dataset corpus we use [17] is one such composite, drawing legitimate samples from Enron/SpamAssassin and phishing samples from the Nazario archive; its license is LGPL-3.0 and its labels are binary, Safe Email/Phishing Email.

2.2. Classical Machine Learning Pipelines

Fette et al. [18] introduced one of the first systematic feature-engineered phishing email classifiers, combining nine hand-crafted features with a Random Forest. Toolan and Carthy [19] expanded this set to 40 features and reported feature-importance rankings. Hamid and Abawajy [20] explored header-level features. Bagui et al. [21] reported a Random Forest baseline on a 6000-email balanced corpus. Earlier work has emphasised hand-engineered features over simple bags-of-words. We show that this conclusion depends strongly on vocabulary size and representation.

2.3. Transformer and Deep Learning Pipelines

Recent works apply pre-trained language models to phishing and spam email classification. Salloum et al. [9] survey NLP and transformer techniques for phishing email detection, with reported F1 around 0.96. Tida and Hsu [10] use transfer learning with a BERT model for message classification. Fang et al. [11] combine convolutional and recurrent (BiLSTM) layers with an attention mechanism over multilevel email embeddings. None of these report a CPU latency budget, and all were trained and deployed on GPUs. Bountakas et al. [22] survey transformer-based phishing detection more broadly.

2.4. Class Imbalance: Methods and Benchmarks

He and Garcia [23] and Krawczyk [24] survey the imbalanced-learning field. Galar et al. [25] specifically benchmark ensembles for imbalanced data. SMOTE [26] and its descendants Borderline-SMOTE [27] and ADASYN [28] are the dominant *data-level* techniques; cost-sensitive learning and ensemble re-weighting [29,30] are the dominant *algorithm-level* techniques. Lemaître et al. [31] provide the imbalanced-learn library we use here. The literature has repeatedly observed that SMOTE-style techniques are *more useful as the imbalance ratio grows*; for moderate IR (<2), the gain over class-weighting is often within noise.

2.5. Phishing and Human-Facing Operational Security

IoT and WSN security research traditionally emphasises protocol-level and infrastructure-level threats, including routing manipulation, node compromise, jamming, sinkhole attacks, Sybil behaviour, wormholes, and anomaly detection over sensor or network traffic [5,6]. These threats are central to sensor-network resilience, but they do not cover the full operational attack surface. Modern IoT/WSN deployments are administered through cloud dashboards, gateway consoles, firmware-management tools, ticketing systems, and email-based workflows [8]. A phishing campaign that compromises an

operator account can therefore become an initial-access path into the sensor-network management plane.

The present study addresses this complementary human-facing layer: it evaluates whether a Calibrated Selective Cascade can provide efficient phishing email detection for low-cost operational infrastructure around IoT/WSN deployments [7,32], where the protected assets are operator and administrator inboxes, gateway-management consoles, and cloud dashboard credentials, and the constraint is the limited inference budget of edge-adjacent operational servers [8].

2.6. Selective Classification and Cascade Architectures

The idea that a low-cost, fast model should handle confident decisions and an expensive slow model should adjudicate the residual is well established outside email security. Selective classification with a reject option dates back to Chow's rule [33] and has been formalised as a calibrated-confidence decision problem; deep selective classifiers extend the idea with learned abstention heads [34]. Two-stage cascades are the workhorse of latency-sensitive computer vision (Viola–Jones boosted cascades [35]) and modern model-serving systems (BranchyNet early-exit [36]). In NLP, they have been applied to LLM cost optimisation (FrugalGPT [37]). The phishing detection literature has not formulated a cascade that fuses a Platt-calibrated low-cost classical classifier, including both a feature-engineered arm and a high-vocabulary sparse-linear arm, with a transformer encoder under a validation-tuned probability-band deferral rule; existing two-stage phishing pipelines (e.g., header-based pre-filtering followed by content classification) are heuristic and not calibrated against a deferral budget.

2.7. Position Relative to Prior Work

Cascade architectures, selective classification, SMOTE-plus-bagging pipelines [26,38], TF-IDF [39], and classical-vs-transformer comparisons on a single corpus are all established in prior work; quarterly threat statistics for the broader phishing landscape are reported by APWG [40]. The contribution of this paper is to situate that body of work within a multi-corpus phishing email benchmark: (i) a fully crossed 231-configuration matrix selects the low-cost arm on a public corpus with held-out evaluation; (ii) the cascade is formulated as a Platt-calibrated probability-band deferral policy and tuned only on a validation partition; (iii) the empirical comparison spans four public corpora; (iv) the per-source decomposition of the multi-source Dataset C exposes within-source heterogeneity that single-number benchmarks conceal; and (v) the feature set/balancer hierarchy is tested with a Friedman omnibus + Nemenyi post hoc rather than pairwise comparisons.

3. Materials and Methods

3.1. Datasets and Preprocessing

3.1.1. Datasets

We use four publicly available phishing email corpora distributed on the Hugging Face Hub. Each is deduplicated against all corpora processed before it by a 200-character prefix match.

Dataset A (primary). zefang-liu/phishing-email-dataset [17]. After dropping rows with missing email bodies, the corpus contains 18,634 emails labelled Safe Email ($n = 11,322$; 60.76%) and Phishing Email ($n = 7312$; 39.24%); IR = 1.55. It mixes legitimate email from Enron and SpamAssassin with phishing from the Nazario archive [13–15].

Dataset B (external). jmmrcp/phishing-IBM [41]. Raw form, 16,672 emails; after 200-character prefix dedup against A (30.99% overlap removed), 11,505 emails (legitimate $n = 5809$, phishing $n = 5696$; IR = 1.02).

Dataset C (multi-source). puyang2025/seven-phishing-email-datasets [42]. A 2025 release that normalises seven historically distinct corpora into a single schema (TREC-05, TREC-06, TREC-07, CEAS-08, Enron, SpamAssassin, Ling-Spam) with a `dataset_name` column, identifying each row's source. Raw form, 203,017 rows; after dedup against A and B (0.36% removed), we draw a stratified subsample of 29,996 emails, stratified by `dataset_name` so that the per-source composition is preserved (TREC-05: 8169; TREC-07: 7987; CEAS-08: 5818; Enron: 4409; TREC-06: 2433; SpamAssassin: 765; Ling-Spam: 415). Class composition: legitimate $n = 15,934$; phishing $n = 14,062$; IR = 1.13.

Dataset D (modern compilation). drorrabin/phishing_emails-data [43]. A 2024 release used for LLM fine-tuning; each row in the raw form is wrapped in an instruction prompt that we strip with a regex to recover the underlying body. After dedup against A and B (0% overlap) and a ≥ 16 -character minimum-length filter: 30,651 emails (legitimate $n = 16,842$, phishing $n = 13,809$; IR = 1.22).

The four corpora together cover 90,786 emails with no detectable cross-corpus duplicates above our 200-character prefix threshold (Figure 1). These datasets evaluate the email security layer relevant to IoT/WSN operations; they are not sensor-network traffic datasets, and we do not relabel email corpora as WSN data. All experiments were implemented in Python 3.13 with open-source libraries: scikit-learn 1.7.2, imbalanced-learn 0.14.0, PyTorch 2.11.0, Hugging Face Transformers 5.7.0 and Datasets 4.8.5, sentence-transformers 5.4.1, XGBoost 3.2.0, LightGBM 4.6.0, NLTK 3.9.2, NumPy 2.3.5, pandas 2.3.3, SciPy 1.16.3, and Matplotlib 3.10.1/Seaborn 0.14.1.

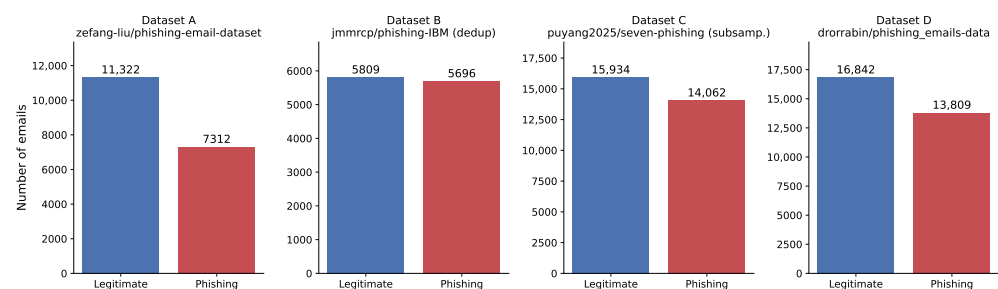


Figure 1. Class distribution of the experimental corpus used for all training and evaluation in this paper (Dataset A: 18,634; B: 11,505; C: 29,996 stratified subsample; D: 30,651). Counts are after removing rows with empty bodies, and for B, C, and D, the cross-corpus prefix duplicates against the corpora processed before them; Dataset D additionally applies a 16-character minimum-length filter. The stricter within-corpus cleaning audit of Table 1 is diagnostic, and the two count sets are reconciled at the end of Section 3.1.

Table 1. Per-corpus cleaning and label-conflict audit. Near-duplicates are MinHash-estimated from a sample of up to 8000 rows per corpus and scaled back to corpus size. Dataset C's manuscript-level count (29,996) is a stratified subsample (by `dataset_name`) drawn from the experimental corpus after cross-corpus deduplication against A and B, not from the stricter within-corpus audit-cleaned pool reported in this table; the two cleaning levels are reconciled at the end of Section 3.1.

DS	Raw	Miss.	Short	Exact	Prefix	X-Corp	Near	Lbl Conf.	Strict Audit-Cleaned Pool
A	18,650	16	545	563	614	0	207	0	16,912
B	16,672	0	22	0	704	4604	715	0	11,342
C	203,017	0	414	5	21,069	695	1288	0	180,834
D	30,651	0	0	0	6	0	0	0	30,645

3.1.2. Dataset Cleaning and Label-Conflict Audit

To verify that the four corpora are clean of artefacts commonly observed in public phishing email collections, we ran a per-corpus audit recording raw row count, rows with

missing bodies, rows shorter than 16 characters, exact duplicates, 200-character prefix duplicates within the corpus, cross-corpus prefix duplicates, MinHash near-duplicates (Jaccard ≥ 0.85 on 5-character shingles, sampled to up to 8000 rows per corpus and scaled back to corpus size), and same-text, different-label conflicts checked *after* lowercasing and whitespace normalisation (Table 1). The 200-character prefix check is deliberately conservative: it identifies messages that share an identical opening even when the body length, signature, or trailing salutation differs, which is the typical shape of templated phishing kits. The MinHash sweep complements this by catching paraphrased near-duplicates that share the bulk of their 5-shingle vocabulary but not their opening 200 characters; in practice, these two checks remove largely overlapping populations and the residual MinHash near-duplicate estimates (Table 1, *Near* column) are small enough relative to the corpus size that any leakage into a stratified 80/20 split is below the within-split sampling variability we report from 5-fold CV.

Two cleaning levels reconcile the counts in Figure 1 and Table 1. The experimental corpus used throughout this paper removes null bodies. For B, C, and D, it also removes cross-corpus prefix duplicates against the corpora processed before them. Dataset A is processed first, so this leaves 18,634 messages, with the count shown in Figure 1. The audit then quantifies within-corpus exact-, prefix-, and short-row removals. For Dataset A, this gives a stricter pool of 16,912 messages (Table 1: raw 18,650, minus 16 missing, 545 short, 563 exact, and 614 prefix duplicates). We re-ran the recommended high-vocabulary SVM on all four corpora after removing exact and 200-character-prefix duplicates and rows shorter than 16 characters. The held-out F1 changes are small in every case: Dataset A 0.961 to 0.976, B 0.990 to 0.983, C 0.958 to 0.959, and D 0.999 to 0.999. The largest change is 0.015, and the change on Dataset C, which carries the largest raw duplicate count, is only 0.001. The hand-engineered pipeline behaves the same way on Dataset A (held-out F1 0.839 to 0.833). Removing within-corpus duplicates therefore does not inflate the reported results on any corpus. Table 1 is a diagnostic cleaning audit, not the sample-size table used for the main experiments; the experimental counts are those shown in Figure 1.

3.1.3. Dataset D Wrapper-Removal Audit

Dataset D wraps each email in an instruction prompt. We strip this wrapper with a regular expression before featurisation. Of the 30,651 raw rows, 30,651 are wrapped; the stripper successfully extracts the body from all of them; a 20-row manual-check sample confirms 100% extraction accuracy.

3.1.4. Text Normalisation

Each email body is lowercased, tokenised with NLTK, restricted to alphanumeric tokens, filtered against the NLTK English stop-word list, and reduced to stems with the Porter algorithm. The normalised string is the input to the TF-IDF vectoriser.

3.2. Feature Engineering and the Low-Cost Arm

3.2.1. Feature Extraction

We construct three nested feature sets.

Compact interpretability baseline (5 columns).

A `TfidfVectorizer` with `max_features = 5` and `ngram_range = (1, 2)` is prepared. This is *not* this paper's strongest lexical baseline; it exists to keep the dimensional count of the text channel comparable to the ten hand-engineered features (15 total in the combined set vs. 5 here), so that the 231-configuration matrix isolates the feature-family effect from a confounded vocabulary-size effect. The accuracy-oriented lexical baselines on each corpus are the six high-vocabulary TF-IDF configurations evaluated separately in Section 4.2.3

(word 5000–10,000-feature TF-IDF with Linear SVM or Logistic Regression, including word-bigram and character- n -gram variants), one of which matches the CPU-feasible fine-tuned DistilBERT reference on Dataset A. We additionally sweep `max_features` $\in \{5, 100, 1000, 5000, 10,000\}$ in Section 4.2.3 to show how the feature set effect shrinks as the vocabulary grows.

Custom-only (10 columns).

Ten hand-engineered features grounded in the phishing email literature are prepared [18,19]: `has_phishing_keywords` (binary indicator of any of {urgent, verify, account, password, click here, update, security alert, login, confirm, suspicious activity, invoice, payment, bank, credit card, personal information, act now, limited time, winner, congratulations, ssn}); `has_links` and `num_links` (URL presence and count); `has_url_shortener` (binary, for any of {bit.ly, tinyurl, goo.gl, t.co, ow.ly, is.gd, buff.ly}); `has_html_form` (binary, presence of `<form>`); `uppercase_ratio` (fraction of alphabetic characters that are uppercase); `num_exclamation` (count of '!'); and `num_characters`, `num_words`, `num_sentences` (length statistics). This set is small and chosen by design rather than by search. It covers the three levers a phishing message must pull to induce a click: authority impersonation (`has_phishing_keywords`), urgency (`num_exclamation`, `uppercase_ratio`), and an actionable element (`has_links`, `num_links`, `has_url_shortener`, `has_html_form`). The three length statistics separate theatrical phishing from verbose business mail. Section 5.3 develops the rationale. Because this study intentionally restricts the input to email body text, header-, sender-reputation-, URL-reputation-, attachment-, and authentication-derived signals are outside the evaluated feature space. This set is therefore not the most discriminative feature set possible. The high-vocabulary lexical baselines in Section 4.2.3 provide a stronger, less hand-curated alternative. The leave-one-out analysis in Section 4.1.5 quantifies the contribution of each retained feature. Each feature is auditable: an analyst inspecting a prediction can be shown *which* feature is fired with *which* value.

Combined (15 columns).

This comprises a concatenation of text-only and custom-only features. The 15-feature hand-engineered pipeline is referred to as the *compact pipeline* and serves as one low-cost-arm variant in CSC.

3.2.2. Imbalance-Handling Strategies

We compare seven strategies, applied *only to the training fold* to prevent synthetic-sample leakage: (1) None; (2) Class-weighting (`class_weight='balanced'` on classifiers that support it); (3) Random Under-Sampling (RUS); (4) SMOTE [26]; (5) Borderline-SMOTE [27]; (6) ADASYN [28]; and (7) SMOTE-Tomek [44] (SMOTE followed by Tomek-link cleaning).

3.2.3. Classifiers

We evaluate eleven classifiers: Logistic Regression, Linear SVM (calibrated for probability output via `CalibratedClassifierCV`), Gaussian Naive Bayes, Decision Tree (max depth 10), AdaBoost (200 estimators), Gradient Boosting (200), Random Forest (200), Extra-Trees (200), Bagging (200), XGBoost (400, depth 6, lr 0.1) [45], and LightGBM (400, lr 0.1) [46]. Hyperparameters are fixed at library-typical values; we deliberately do not perform per-classifier grid search to isolate the imbalance-handling $\times$ feature set interaction.

3.3. Transformer Arm and Calibration

3.3.1. Transformer Baselines

Rationale. The cascade is targeted at deployment regimes where transformer inference is the binding cost (per-email latency, encoder weights on disk, retraining time). Frozen MiniLM and frozen DistilBERT mean-pool with a linear head are the most common production-grade transformer setups for moderate-budget CPU deployments and are therefore the relevant *deployment-equivalent* baselines for the cascade. A fine-tuned DistilBERT is additionally included as an upper-bound reference (Section 4.2.4); this result (union-trained, mean F1 = 0.983 across the four corpora) is used as an upper-bound reference rather than as the main deployment-equivalent comparator.

MiniLM (frozen, lower-cost reference arm): We embed each body with sentence-transformers/all-MiniLM-L6-v2 (22.7 M parameters, 384-dim L2-normalised embeddings) on a CPU [47] and train Logistic Regression and Bagging on top with and without SMOTE. Bodies longer than 2000 characters are truncated. *DistilBERT (frozen mean-pool, higher-accuracy reference arm):* We mean-pool (mask-aware) the last hidden state of distilbert-base-uncased [48] (66 M parameters, 265 MB on disk, 768-dim embeddings), batch size 16, max sequence length 256, with bodies truncated to the first 1500 characters before tokenisation; the downstream classifier is Logistic Regression with and without SMOTE. Both baselines share the held-out test split with the compact pipeline.

Latency-measurement environment. Per-email latency is measured on a 2-core x86-64 CPU (Intel-class, hyperthreading disabled), 16 GB RAM, Ubuntu 22.04 inside WSL2 on Windows 11, Python 3.11, PyTorch 2.2 (CPU build), sentence-transformers 2.7.0, transformers 4.41, scikit-learn 1.4, single-thread NumPy/MKL. Each model is warmed up with three predictions before timing; reported figures are mean wall-clock time over 100 single-email predictions on the held-out test partition, with feature extraction (NLTK tokenisation + Porter stemming for the low-cost arm; tokeniser + encoder pass for the transformer arms) included in the timed block. We deliberately measure single-email serial latency rather than batched throughput because the deployment regime motivating the cascade is in-mailbox/gateway scoring of individual messages; gateways that batch incoming mail can amortise the transformer's encoder cost, and the ratio we report is therefore an *upper bound* on the transformer's relative latency disadvantage. We do not report p50/p95/p99 latency distributions because single-email Python-level overhead dominates the spread on this hardware. Tail-latency percentiles should be evaluated in future work under a production inference runtime. Because latency was measured on x86-64/WSL2 rather than ARM-class edge hardware, the reported milliseconds should be interpreted as relative model cost estimates rather than deployment-certified gateway latency. We therefore use latency only as a relative quantity. It captures the ratio between classical and transformer inference and the convex-combination latency of the cascade. On-device ARM/edge benchmarking is left as future work.

3.3.2. Calibration Analysis

We measure calibration of the compact pipeline with the Brier score [49] and the Expected Calibration Error (ECE) [50] over 15 equal-width probability bins. We then re-train the pipeline inside CalibratedClassifierCV with sigmoid (Platt [51]) scaling and report the post-calibration Brier and ECE.

3.4. Design of the Calibrated Selective Cascade

Deployment model. The target deployment is a gateway-adjacent or SOC-side classifier in IoT/WSN operational environments, where incoming emails to administrators, gateway operators, or security teams are scored before user interaction or triage. The low-cost arm

handles most messages with a fast, inspectable decision, while the transformer arm is invoked only for messages whose calibrated probability falls inside the uncertainty band. This places the heavier computation where it is most useful, which matters when the host is a low-cost edge server rather than a datacentre node.

Design principle: Composition over novelty. CSC deliberately reuses two well-understood, independently trained classifiers and adds value through a per-email policy that decides which one acts on each message. Building the routing layer on existing arms keeps every component inspectable, lets each arm be retrained or swapped independently, and concentrates the design contribution in the calibrated deferral rule rather than in a new learning algorithm. CSC follows the selective-classification family and instantiates it for calibrated phishing email routing under deployment constraints; the deferral rule is in the same class as Chow’s reject rule and confidence-based early-exit routing [33,34,36]. The adaptation is specific to this deployment setting. The low-cost arm is Platt-calibrated, so the deferral band lives on a meaningful probability axis and the deferral rate reads directly as a drift signal. The band is tuned only on a validation partition and reported once on a disjoint test partition, not tuned on the evaluation data. And CSC is instantiated classical-first (Section 4.5): its high-vocabulary sparse-linear arm already matches the transformer in-domain, so the cascade is a cost, interpretability, and monitoring layer rather than an accuracy mechanism.

Formally, CSC is a per-email deferral policy that composes a calibrated low-cost arm with a transformer arm. The low-cost arm can be either of two classifiers, and defines two cascade families. *CSC-HE* uses the 15-feature hand-engineered pipeline (TF-IDF + 10 custom + SMOTE + Bagging, Platt-calibrated; Sections 3.2 and 3.3.2). It is fully per-feature auditable and is retained as an interpretable diagnostic cascade. *CSC-SVM* uses the Platt-calibrated high-vocabulary Linear SVM (Section 4.2.3). It is the recommended deployment variant. Let $p_L(x) \in [0, 1]$ denote the calibrated low-cost-arm probability for email x and $p_D(x)$ the transformer-arm probability. We define two threshold parameters $\theta_l \leq 0.5 \leq \theta_h$ and the cascade decision rule

$$\hat{p}_{\text{CSC}}(x) = \begin{cases} p_D(x), & \theta_l \leq p_L(x) \leq \theta_h \\ p_L(x), & \text{otherwise} \end{cases} \quad (1)$$

Emails with calibrated probability inside $[\theta_l, \theta_h]$ are treated as uncertain and deferred to the transformer arm; emails outside the band retain the low-cost-arm decision. The deferral band is selected on the validation partition only and is then applied unchanged to the held-out test partition. The deferral rate $\delta = \Pr[\theta_l \leq p_L(X) \leq \theta_h]$ is the cascade’s single tunable trade-off parameter: at $\delta = 0$, the cascade reduces to the low-cost arm; at $\delta = 1$, it reduces to the transformer arm. Effective per-email latency is the convex combination $(1 - \delta)L_L + \delta L_D$, where $L_L = 8.0$ ms is the low-cost-arm latency, tokenisation-dominated and comparable for both classical arms, and $L_D \in \{18.5, 35.5\}$ ms is that for MiniLM/DistilBERT, respectively. The band $[\theta_l, \theta_h]$ is tuned on a 2.5%-step grid over the unit interval (≈ 800 candidate bands), and named operating points are reported per cascade variant: *Budget-10* and *Budget-20* (max F1 subject to $\delta \leq 0.10$ and $\delta \leq 0.20$, respectively); *Balanced* (max F1 $- 0.1\delta$, one F1 point per 10% deferral); *Latency-15* (max F1 targeted at a 15 ms effective-latency budget); and *Aggressive* (max F1 irrespective of δ). The main cascade results use the validation-tuned protocol of Section 4.3.2 and report the five points Budget-10, Budget-20, Balanced, Latency-15, and Aggressive; the F1-latency Pareto summary reports the Balanced and Aggressive operating points per cascade variant. The hand-engineered family is reported as CSC-HE-MiniLM and CSC-HE-DistilBERT (Sections 4.3 and 4.3.2); the recommended CSC-SVM-DistilBERT is analysed in Section 4.5.3. The cascade is implemented as a simple

routing policy on top of the two already-trained arms and adds no training overhead. Figure 2 shows the routing schematic.

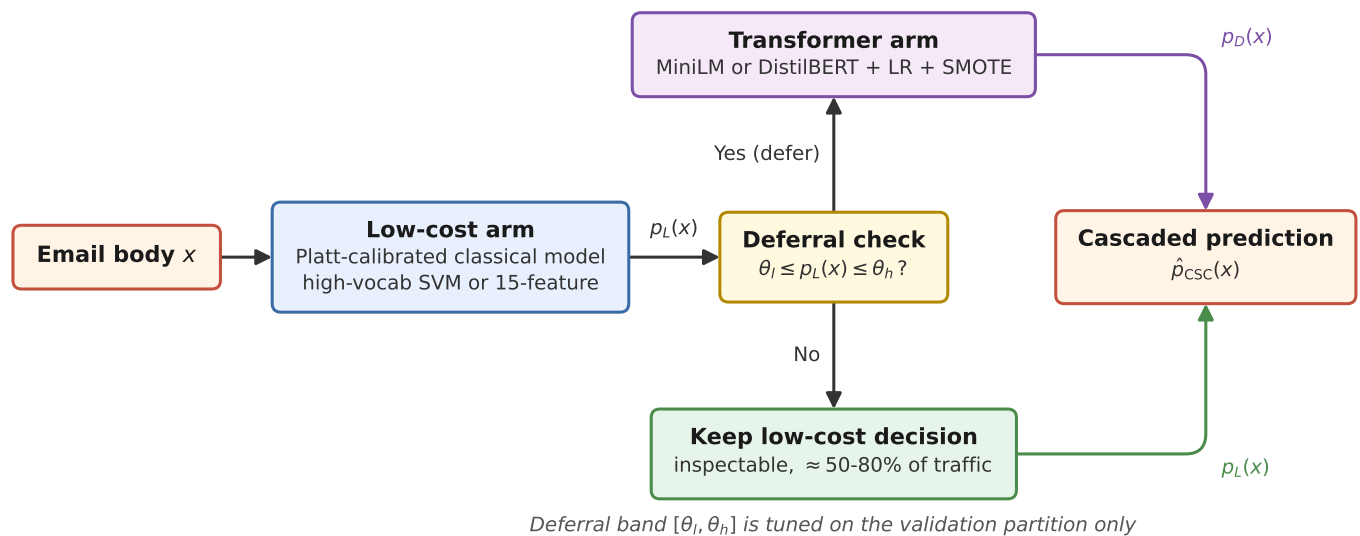


Figure 2. The Calibrated Selective Cascade routing policy for resource-constrained operational email security. In IoT/WSN environments, this classifier can be deployed at a mail gateway, SOC triage layer, or gateway-adjacent operational server. Each email is scored by a Platt-calibrated low-cost classifier; emails whose calibrated probability falls inside the validation-tuned deferral band $[\theta_l, \theta_h]$ are forwarded to a transformer head, otherwise the low-cost decision is retained.

3.5. Evaluation Protocol and Statistical Analysis

3.5.1. Evaluation Protocol

For reproducibility, every experiment in this paper fixes `random_state = 42` for all data splits, resamplers, and classifiers, so the entire pipeline (preprocessing, feature extraction, training, cascade tuning, and statistical testing) is deterministic; the full reproducibility statement and artefact list are given in the Reproducibility subsection at the end of this section. Every (`feature_set` $\times$ `balancer` $\times$ `classifier`) cell is evaluated under two protocols: (a) 5-fold stratified CV on the 80% training partition with resampling inside each training fold; (b) a held-out test on a 20% stratified test partition, touched once per cell. Configuration selection uses the validation signal only: five-fold CV for the compact feature matrix, and a held-out validation partition for the high-vocabulary arm (Section 4.2.3). The test partition is used for final reporting, not for choosing among configurations. Three splitting conventions appear in this paper, each tied to a different selection task: (i) an 80/20 train/test split for the per-cell metrics of the configuration matrix, with selection by five-fold CV inside the 80% training partition; (ii) a 60/20/20 train/validation/test split for choosing the high-vocabulary arm; and (iii) a 50/50 validation/test split of the held-out partition for tuning the cascade band after both arms are fixed. In all three, the final test partition is reported once and never used for selection. We report accuracy, balanced accuracy, precision (phishing), recall (phishing), F1 (phishing), ROC-AUC, AUPRC, and the Matthews correlation coefficient (MCC) [52]. For moderate imbalance, F1 and MCC together provide a faithful summary of minority-class performance; accuracy in isolation is well known to be misleading [23]. The statistical tests for head-to-head comparisons are described in Section 3.5.2.

3.5.2. Statistical Analysis

For the head-to-head comparison of the compact pipeline against weaker baselines, we report *three* complementary statistical quantities. First, a paired bootstrap test on the held-

out predictions is conducted: 1000 resamples of the test set with replacement, recompute F1 on each resample for both pipelines, and report the 95% percentile confidence interval on the paired F1 gap. Bootstrap is the right tool for a single held-out partition because it captures sampling variability in the test set without re-fitting the models. Second, a 5×2 -fold cross-validated paired test is carried out [53] (5 random repetitions of a 50/50 stratified split, two folds each): we report the mean and standard deviation of the 10 paired F1 differences, the Wilcoxon signed-rank statistic (non-parametric, robust to non-Gaussian distributions), and the paired t -statistic (parametric). The 5×2 cv protocol controls for the dependence between the train and test partitions and is the standard recommendation when reporting whether one classifier is reliably better than another. Third, we report Cohen's d as the effect size; values above 0.8 are conventionally considered large.

3.5.3. Friedman + Nemenyi Omnibus Tests

Beyond the pairwise paired bootstrap and 5×2 cv tests, we report a Friedman omnibus test [54] across all 11 classifiers as “datasets” and the 7 balancers as “treatments” (with held-out F1, AUC, and MCC on the combined feature set as the response), and a symmetric test treating the 3 feature sets as treatments and all 77 (classifier $\times$ balancer) cells as datasets. If Friedman rejects, we apply the Nemenyi pairwise post hoc at $\alpha = 0.05$ with the corresponding critical-difference threshold.

3.5.4. Multi-Corpus Protocol

The low-cost pipeline (SMOTE + combined features + Bagging) is evaluated on each of the four corpora under the in-domain (80/20 stratified split) and 4×4 cross-transfer protocols (16 train $\rightarrow$ test cells plus a union row A+B+C+D $\rightarrow$ each held-out). Dataset C is additionally evaluated per-source (the 7 contributing corpora) under two protocols: *within-C* (trained on 80% of C, evaluated per source on the held-out 20%) and *external* (trained on the union of A+B+D training partitions, never seeing any C row, evaluated on each C source).

3.5.5. Operating-Threshold Analysis

We sweep the decision threshold across $\{0.30, 0.40, 0.45, 0.50, 0.55, 0.60, 0.70, 0.80, 0.90\}$ on the held-out probabilities of the compact pipeline and report precision, recall, F1, raw false-positive count, raw false-negative count, false-positives per 10,000 legitimate emails, and false-negatives per 10,000 phishing emails. We additionally report recall at $\geq 95\%$ precision and precision at $\geq 95\%$ recall.

3.5.6. Adversarial Transformations

We apply five attacker-side transformations to only the phishing class of A's held-out test set, re-extract features through the production pipeline, and re-score: `remove_exclamation` (strip every '!'); `lengthen_filler` (append four copies of a 33-token business-style filler); `remove_keywords` (regex-strip the 15 most frequent phishing-keyword tokens); `unshorten_urls` (rewrite any common URL shortener to `example.org`); `lowercase_all`; and `combined` (`remove_exclamation` + `lengthen_filler` + `remove_keywords` + `unshorten_urls` + `lowercase_all`). Transformations are applied to test bodies only (the model is not retrained). *These transformations are feature-space stress tests that identify which human-editable stylistic cues the model depends on.* Each transformation maps to a low-cost real-world attacker strategy (Section 4.7.1); a sophisticated adversary with classifier-query access could craft stronger evasions, which we leave to future work.

3.5.7. Reproducibility

All random seeds are fixed (`random_state = 42`). The full pipeline (preprocessing, feature extraction, model training, cascade tuning, statistical testing, and figure generation) is scripted end-to-end with fixed seeds and per-corpus feature matrices. The code, intermediate feature files, trained models, and cascade band sweep are available from the corresponding author upon reasonable request.

4. Results

The main cascade result is summarised in Section 4.3.2. The protocol is a strict validation-only thresholding across all four corpora. The held-out test F1 ranges from 0.878 (Budget-10 on A) to 0.992 (Aggressive on D). With the hand-engineered low-cost arm, the CSC-HE-DistilBERT Aggressive point reaches $F1 = 0.940$ on Dataset A at 22.3 ms effective per-email CPU latency, recovering most of the low-cost-to-DistilBERT gap. The Budget-20 point reaches 0.905 at 13.6 ms, with roughly 80% of decisions kept on the auditable low-cost arm. The size of this gap depends on how weak the low-cost arm is. With a strong classical low-cost arm, the cascade adds little, as Section 4.5 shows. The rest of this section presents the supporting evidence. Section 4.1 selects the low-cost arm and quantifies the effect of imbalance handling. Section 4.2 evaluates the transformer and high-vocabulary lexical baselines. Sections 4.3 and 4.3.2 analyse the F1–latency trade-off of CSC, and Section 4.5 stress-tests the strong classical baseline and the classical-first cascade. Section 4.4 examines cross-corpus generalisation. Sections 4.6 and 4.7 analyse calibration, decision thresholds, stylistic stress tests, and error patterns. Sections 4.3 and 5.6 summarise the deployment outcome; the remaining subsections justify the component choices and analyse the boundary conditions.

4.1. Selecting the Low-Cost Arm

4.1.1. Baseline Classifiers on Raw Imbalanced Data

Trained on the raw 60/40 corpus with the text-only TF-IDF features, Logistic Regression and Linear SVM achieve a held-out F1 of 0.534 with phishing-class recall of 0.523. The two linear classifiers, in this configuration, are little better than random on the minority class, a result well known from the literature [23,24] and reproduced here as a baseline check. Gaussian Naive Bayes inverts the failure mode (recall is high but precision is very low). The tree ensembles fare slightly better but still cluster around $F1 \approx 0.60$ and $AUC \approx 0.74$, improved but well below the transformer and cascade operating points reported later in this section.

4.1.2. The 231-Configuration Matrix

We evaluate the held-out F1 of every (balancer $\times$ classifier) cell under the combined feature set. The dominant pattern is that the strongest ensemble classifiers—Random Forest, Extra-Trees, Bagging, XGBoost, and LightGBM—cluster near $F1 = 0.84$, with the ten best configurations spanning only 0.0035 F1. Gradient Boosting performs below this leading cluster but remains above the single Decision Tree and the linear/Gaussian baselines. Linear classifiers and Gaussian Naive Bayes lag by 10–16 F1 points in this low-vocabulary setting. The maximum spread within any tree-ensemble row across balancers is ≤ 0.010 F1. Within this leading cluster, we use Bagging for the compact arm: it sits under 0.0035 F1 from the best ensemble, and its averaged bootstrap estimates give stable Platt-calibrated probabilities for the deferral band. The choice is not accuracy-critical since the recommended low-cost arm is the high-vocabulary SVM rather than any tree ensemble (Section 4.2.3); substituting Extra-Trees, the best combined-feature ensemble, changes the compact arm's F1 by less than 0.01.

4.1.3. Effect of Feature Representation Relative to Imbalance Handling

Figure 3 isolates the effect of the feature set. With SMOTE applied, the best classifier under the text-only set ($\text{max_features} = 5$) reaches $F1 = 0.683$ (XGBoost); under the custom-only set, $F1 = 0.775$ (Extra-Trees); and under the combined set, $F1 = 0.847$ (Extra-Trees). The gap text-only $\rightarrow$ custom-only is $+0.092$ F1; custom-only $\rightarrow$ combined is $+0.072$ F1; and total text-only $\rightarrow$ combined is $+0.164$ F1. Note that the 5-feature TF-IDF baseline is deliberately small for interpretability parity with the 10 hand-engineered features; Section 4.2.3 relaxes this and shows that the qualitative ordering (feature set effect $\gg$ balancer effect) is preserved as the vocabulary grows.

For comparison, the largest gap between any two imbalance-handling strategies *within the combined feature set* is only $+0.016$ F1 (mean SMOTE = 0.786 vs. mean none = 0.770). The effect of feature representation is therefore an order of magnitude larger than the effect of balancer choice on this corpus.

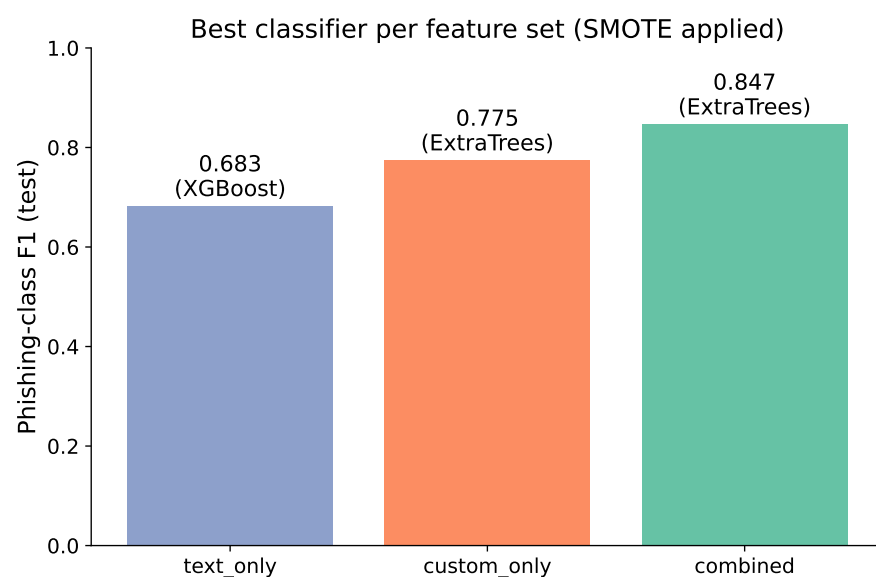


Figure 3. Best classifier per feature set (SMOTE applied). The 9-point gap text-only $\rightarrow$ custom-only and the further 7-point gap custom-only $\rightarrow$ combined are substantially larger than the 1.6-point gap among balancers.

4.1.4. Statistical Significance

A paired bootstrap test (1,000 resamples) on the held-out test predictions of the compact pipeline (SMOTE + combined features + Bagging) versus a no-balancing + text-only baseline returns an F1 gap of $+0.242$ (95% CI $[+0.221, +0.266]$, $p < 0.001$). This gap is well outside the bootstrap variability. Holding the feature set fixed at combined, the SMOTE-vs-no-balance 95% CI is narrower than 0.01 and includes zero. Re-balancing alone does not significantly improve the model once the features carry enough signal.

A 5×2 -fold cross-validated paired test [53] (10 paired F1 measurements) confirms the proposed-vs-text-baseline gap is not driven by the choice of train/test split. The mean $\Delta F1$ is $+0.178$ (std 0.038); Wilcoxon signed-rank $W = 55$ with $p = 0.001$; paired t -test $t = 14.86$ with $p = 6 \times 10^{-8}$; and Cohen's $d = 4.70$. Both non-parametric and parametric tests reject the null at $p < 0.005$.

The Friedman omnibus test (Section 3.5.3) rejects equality of feature set means at $\chi^2 = 100.3$, $p < 10^{-22}$ ($CD_{0.05} = 0.38$), but for balancers, Friedman returns $\chi^2 = 11.7$, $p = 0.068$ on MCC—not significant; the omnibus therefore confirms “feature representation $\gg$ imbalance handling” at the statistical significance level: every pairwise feature set rank difference exceeds the critical difference, while the balancer ranks do not separate.

4.1.5. Leave-One-Out Feature Importance

We retrain the best configuration (SMOTE + Bagging + combined features) with each of the ten custom features dropped in turn (Table 2). The single most discriminative feature is the count of exclamation marks—a stylistic indicator of urgency that phishing emails inherit from their template generators. The three message length features contribute approximately the same: phishing emails in this corpus are systematically shorter than legitimate Enron/SpamAssassin emails, the keyword indicator (*has_phishing_keywords*) carries 0.010 F1 of unique signal, and the binary URL features carry essentially none, presumably because their information is largely redundant with *num_links*.

Table 2. Leave-one-out importance of each custom feature. Larger $\Delta F1$ = more important. Baseline F1 (all features) = 0.841. Bold marks the largest $\Delta F1$ (the most important feature).

Feature Removed	F1 Without	$\Delta F1$
<i>num_exclamation</i>	0.807	+0.034
<i>num_words</i>	0.820	+0.021
<i>num_sentences</i>	0.820	+0.021
<i>num_characters</i>	0.820	+0.021
<i>has_phishing_keywords</i>	0.831	+0.010
<i>uppercase_ratio</i>	0.837	+0.004
<i>num_links</i>	0.838	+0.003
<i>has_html_form</i>	0.838	+0.003
<i>has_url_shortener</i>	0.839	+0.002
<i>has_links</i>	0.841	0.000

This ranking is corpus-specific and differs from the link- and keyword-centric emphasis common in parts of the phishing detection literature; on this corpus, *style and length* features dominate. The reason is the corpus composition: legitimate Enron mail is verbose and business-formal, while Nazario phishing samples are short and theatrically punctuated.

4.2. Transformer Baselines and Training Cost

4.2.1. Comparison with Transformer Baselines

Embedding the 18,634-email corpus with *a11-MiniLM-L6-v2* on the CPU is completed in 118.8 s (6.4 ms per email amortised). Mean-pooling DistilBERT’s last hidden state takes 661 s (35.5 ms per email). Table 3 gives the held-out test scores.

Table 3. Compact pipeline vs. two sentence-transformer baselines on an identical 20% held-out test split. *Latency* is per-email inference time on a 2-core CPU; for the transformer pipelines, this is dominated by the encoder pass. The accuracy-oriented high-vocabulary classical model (F1 = 0.961) is reported in Section 4.2.3.

Pipeline	F1	AUC	MCC	Fit (s)	Enc. (ms)	Lat. (ms)	Enc. (MB)
Compact (5 TF-IDF + 10 custom, SMOTE+Bagging)	0.847	0.943	0.746	7.7	—	8.0	—
MiniLM + LogReg	0.929	0.986	0.882	0.1	6.4	18.5	90
MiniLM + LogReg + SMOTE	0.926	0.986	0.876	0.1	6.4	18.5	90
MiniLM + Bagging	0.896	0.973	0.830	1642.7	6.4	≈19	90
MiniLM + Bagging + SMOTE	0.901	0.972	0.837	2089.4	6.4	≈19	90
DistilBERT mean-pool + LogReg	0.947	0.993	0.912	0.6	35.5	35.5	265
DistilBERT mean-pool + LogReg + SMOTE	0.950	0.993	0.917	0.8	35.5	35.5	265

The stronger DistilBERT baseline outperforms the compact pipeline by +0.103 F1 and +0.050 AUC at 265 MB of encoder weights and 35.5 ms of per-email inference time. The cost-realistic MiniLM baseline outperforms it by +0.082 F1 and +0.043 AUC at 90 MB/18.5 ms. Relative to the no-balancing/text-only baseline (F1 = 0.534), the compact pipeline closes

79% of the F1 gap to MiniLM and 75% of the gap to DistilBERT. End-to-end inference is $2.3\times$ faster than MiniLM and $4.4\times$ faster than DistilBERT, and the compact model trains end-to-end in 7.7 s from raw text versus 118.9 s (MiniLM) or 661 s (DistilBERT) for the encoder embedding pass alone. Scoring 10^6 emails on a 2-core CPU takes ≈ 2 h 13 min (compact), ≈ 5 h 09 min (MiniLM), and ≈ 9 h 51 min (DistilBERT). The latency profile is particularly relevant to low-cost operational servers and edge-adjacent IoT/WSN deployments, where full transformer inference for every message may be unnecessary or economically unattractive.

A note on the compact pipeline's latency: 8 ms per single email is dominated not by the classifier itself (Bagging on 15 features predicts in <0.2 ms) but by NLTK tokenisation and Porter stemming on raw text. Neither transformer baseline admits a similar optimisation: the encoder pass is essential.

4.2.2. Training Cost

The compact pipeline trains end-to-end (raw-text feature extraction + SMOTE + Bagging Classifier with 200 estimators, serial workers) in 7.7 s on a 2-core CPU laptop. The MiniLM encoder pass over the full 18,634-email corpus takes 118.8 s and the DistilBERT pass takes 661 s on the same hardware; the downstream linear classifier fit takes <1 s. Whenever new training data arrive, the compact pipeline can therefore be retrained roughly $15\times$ faster than MiniLM and $85\times$ faster than DistilBERT.

Held-out F1 saturates at ≈ 4000 training samples for all three pipelines. Below 1000 samples, the pre-trained transformers retain an advantage over the low-cost pipeline, which motivates the cascade for label-sparse bootstrap deployments.

4.2.3. Competitive High-Vocabulary Classical Baselines

The 231-configuration matrix uses a deliberately small TF-IDF representation for dimensional comparability with the 10 hand-engineered features; it is a controlled baseline rather than the strongest possible TF-IDF baseline. Table 4 reports six high-vocabulary classical configurations on each of the four corpora.

Table 4. Competitive classical baselines with high-vocabulary TF-IDF. F1 on each corpus's held-out test partition.

Configuration	F1 (A)	F1 (B)	F1 (C)	F1 (D)
word TF-IDF 5000 + LR	0.949	0.983	0.949	0.995
word TF-IDF 10,000 + Linear SVM	0.961	0.990	0.958	0.999
word 1–2-g TF-IDF 10k + LR	0.952	0.985	0.953	0.999
char 3–5-g TF-IDF 10k + LinSVM	0.959	0.986	0.952	0.999
word 10k + 10 custom + LR	0.927	0.980	0.934	0.992
word 10k + 10 custom + LinSVM	0.795	0.945	0.960	0.996

The best high-vocabulary classical baseline reaches $F1 = 0.961$ on Dataset A, matching the CPU-feasible fine-tuned DistilBERT reference ($F1 = 0.961$, Section 4.2.4) at orders of magnitude lower training and inference cost (Linear SVM fit time: 0.22 s; pickled size: 0.23 MB). The one low value in Table 4, word 10k + 10 custom features + Linear SVM at $F1 = 0.795$ on Dataset A, is a scaling artefact: concatenating the ten unscaled dense custom features with the sparse, L2-normalised TF-IDF vectors distorts the SVM margin. The Logistic Regression variant (0.927) and the tree ensembles are insensitive to this, so the recommended SVM configuration uses TF-IDF features only. The high-vocabulary classical baseline serves as the recommended low-cost arm of the cascade (Section 4.5) and as the competitive non-transformer reference, while the 15-feature compact pipeline serves as the maximally auditable, per-feature-interpretable arm and as the controlled baseline for the feature-family analysis.

Validation-based selection. The recommended configuration is selected on validation data, not on the test partition. We split each corpus 60/20/20 into train, validation, and test; compare the six high-vocabulary configurations on the validation partition; select the best; refit it on train and validation combined; and report once on the held-out test partition. On all four corpora, the validation partition selects the same configuration, word TF-IDF 10k + Linear SVM. Reported once on the held-out test, it reaches $F1 = 0.960$ on Dataset A, and 0.983, 0.964, and 0.998 on B, C, and D, within 0.002 of the values obtained when the configuration is instead chosen directly on the test data. The hand-engineered arm follows the same principle: its feature set, balancer, and classifier are chosen by five-fold cross-validation on the training partition (Section 3.5). Selected this way, the combined-feature configuration reaches test $F1 = 0.841$, and the feature set ordering (combined > custom-only > text-only) is unchanged from the test ranking. The recommended arm is therefore not an artefact of test set selection.

A paired bootstrap on the Dataset A test partition (2,000 resamples, identical examples for every model) quantifies the comparison with the transformer references. The SVM's $F1$ exceeds the frozen DistilBERT baseline by 0.009 (95% CI [+0.001, +0.017], excluding zero) and is statistically indistinguishable from the fine-tuned reference (−0.002 $F1$, 95% CI [−0.009, +0.005], including zero). The headline claim is therefore that the calibrated SVM beats the frozen transformer and ties the fine-tuned one in-domain, not that it is uniformly superior.

These numbers fall within the range reported by the prior phishing detection literature, which lets us position our results against named methods even though the corpora differ. The transformer line of work reports $F1$ around 0.96–0.97 for BERT-family phishing and spam email detectors [9–11]. Our fine-tuned DistilBERT reaches 0.961 in-domain and a 0.983 union mean (Table 5) in the same band. The classical line of work runs from the feature-engineered Random Forest detectors of Fette et al. [18], Toolan and Carthy [19], and Bagui et al. [21]. We represent it here by the hand-engineered arm and, more competitively, by the high-vocabulary SVM at 0.961. The point of this comparison is not a higher single-corpus $F1$, which is already saturated across methods. It is rather the set of axes that the prior works do not report together: per-email CPU latency and disk cost, a four-corpus cross-transfer matrix, attacker-side robustness, calibration, and behaviour on LLM-style phishing (Sections 4.2–4.5).

Vocabulary size sensitivity. Sweeping $\text{max_features} \in \{5, 100, 1000, 5000, 10,000\}$ (word 1–2-g, Extra-Trees + SMOTE, identical 80/20 split) shows that the five-feature baseline is the worst-case point on the curve, not the typical one: text-only $F1$ rises from 0.671 (5 features) to 0.894 (100) and 0.955 (5000), while the text-only-to-combined gap shrinks from +0.169 $F1$ to +0.005—at a sufficiently large vocabulary, TF-IDF implicitly recovers most of what the hand-engineered features encode (Figure 4). Feature representation therefore dominates in the controlled low-vocabulary comparison, but at high vocabulary sizes, the lexical TF-IDF saturates and the custom-feature gains become marginal (the text-only-to-combined gap is about +0.005 $F1$ at 5000 features and reverses slightly for character n -grams). The balancer effect, by contrast, stays small at every vocabulary size (≤ 0.016 $F1$ within any single configuration). Word-unigram ($F1 = 0.957$ text-only/0.961 combined at 5000 features) and character 3–5-g (0.960/0.959 at 10,000) tokenisations saturate at the same level, so the curve is not specific to the word-bigram choice. This saturation is specific to these legacy-style corpora—high-vocabulary sparse lexical features capture corpus-specific cues (Enron business prose vs. Nazario phishing) that the frozen transformer embeddings do not fully exploit—and does not support a general claim that lexical models dominate transformers for phishing detection; a fine-tuned transformer, a contemporary 2024+ corpus, or LLM-generated phishing text would likely change the ordering.

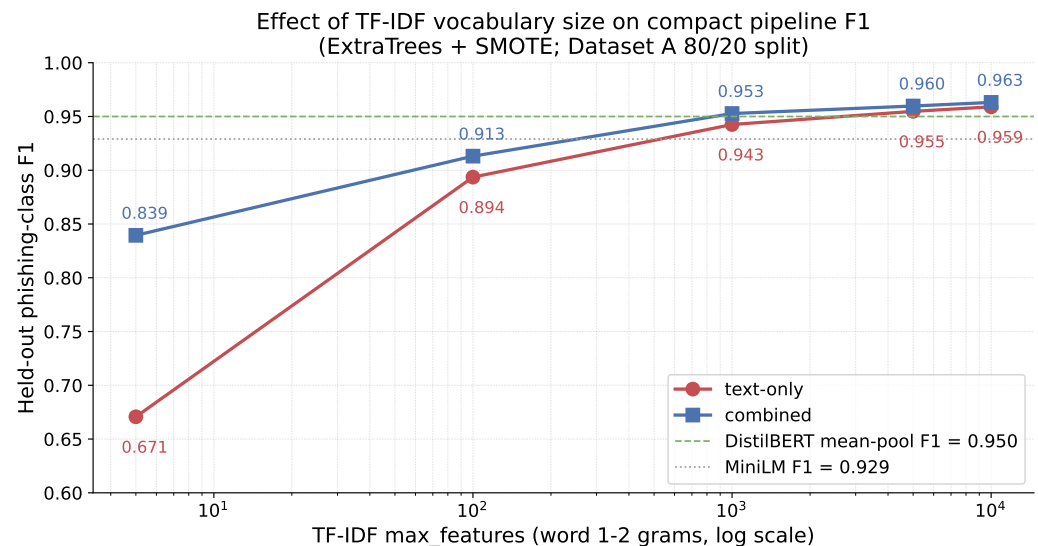


Figure 4. Held-out F1 vs. TF-IDF vocabulary size (word 1–2-g, log scale). Both text-only (red) and combined (blue) curves saturate near F1 = 0.96 at ≥ 5000 features, close to the DistilBERT mean-pool baseline (green dashed line at 0.950) and well above the MiniLM baseline (grey dotted line at 0.929).

Table 5. Fine-tuned DistilBERT reference (CPU-feasible configuration: seq 96, batch 32, 1 epoch, lr $2e-5$, AdamW). Reported as an upper-bound reference for the cascade; when latency and interpretability constraints are removed, fine-tuned transformer-only inference is the appropriate ceiling, and the CSC targets the constrained regime below it.

Pipeline	Test	F1	AUC	MCC	Fit (s)	Lat (ms)	MB
FT DistilBERT (A train)	A	0.961	0.996	0.935	653	11.1	265
FT DistilBERT (union train)	A	0.967	0.997	0.946	4045	9.4	265
FT DistilBERT (union train)	B	0.989	1.000	0.978	4045	9.2	265
FT DistilBERT (union train)	C	0.977	0.998	0.957	4045	9.5	265
FT DistilBERT (union train)	D	0.997	1.000	0.995	4045	9.5	265
Union mean (A,B,C,D)	—	0.983	0.999	0.969	4045	9.4	265

4.2.4. Fine-Tuned DistilBERT (Upper-Bound Reference)

We additionally fine-tune `distilbert-base-uncased` end-to-end with a binary classification head. The CPU-feasible configuration reported here uses max sequence length 96, batch size 32, one epoch, learning rate 2×10^{-5} , and AdamW with 10% warmup. The standard 3-epoch, 256-sequence-length configuration requires a GPU and is documented as a reproducibility reference. Table 5 reports the result on Dataset A. The fine-tuned reference runs at 9–11 ms per email, faster than the 35.5 ms frozen mean-pool baseline, because it encodes a shorter 96-token sequence against the 256 tokens used for the frozen encoder. This is close to the SVM’s 8 ms, so on this hardware, the SVM’s decisive advantages over a fine-tuned transformer are its 0.23 MB footprint (versus 265 MB) and CPU-only training (0.22 s versus GPU minutes), not per-email latency.

4.3. The Calibrated Selective Cascade

4.3.1. Main Result

The CSC (Section 3.4) routes each email through a Platt-calibrated low-cost arm first. Emails whose calibrated probability falls inside a tuned deferral band $[\theta_l, \theta_h]$ go to a transformer arm. Emails outside the band keep the low-cost-arm decision. We report the cascade under the strict validation-tuned protocol of Section 4.3.2: the band is selected on a validation partition and evaluated once on a disjoint test partition. Table 6 gives the five operating points on all four corpora. With the hand-engineered low-cost arm

(F1 = 0.840 on Dataset A), the cascade recovers a large share of the low-cost-to-deep gap. CSC-HE-DistilBERT Aggressive reaches F1 = 0.940 at 22.3 ms, about 0.01 below frozen DistilBERT-only. The Budget-20 point reaches 0.905 at 13.6 ms while keeping roughly 80% of decisions on the auditable low-cost arm. Figure 5 shows the F1–latency frontier across all four corpora.

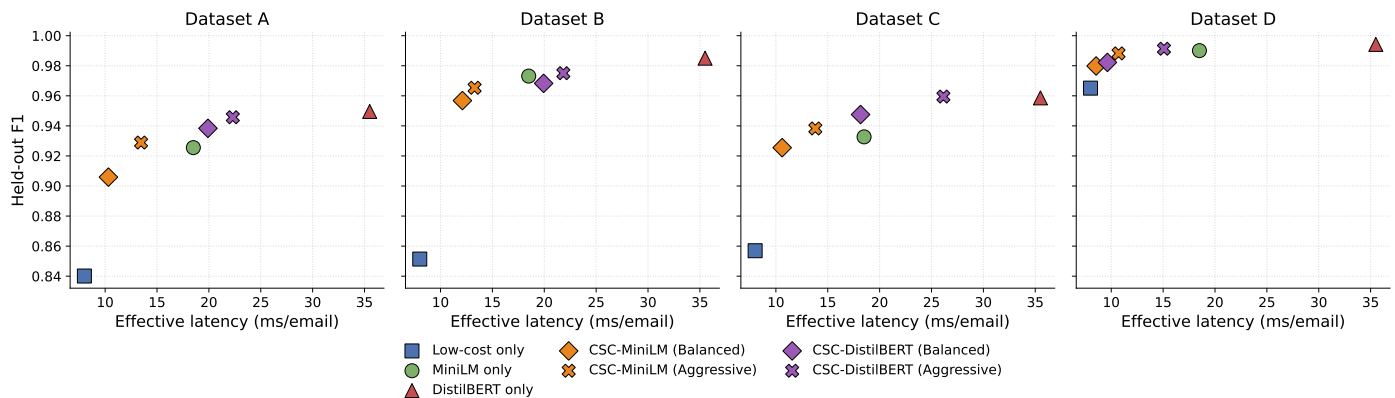


Figure 5. Held-out F1 vs. effective per-email CPU latency. Markers: Three single-model baselines (low-cost, MiniLM, DistilBERT) and two CSC operating points (Balanced and Aggressive) per cascade variant. Each panel is one dataset.

Table 6. Validation-tuned Calibrated Selective Cascade (CSC-HE-DistilBERT). (θ_l, θ_h) chosen on a held-out *validation* partition; *test F1* is reported once on a separate *test* partition. The CSC deferral band was selected only on the validation partition. The held-out test partition was used once for final reporting and was not used to tune θ_l or θ_h .

DS	Op. Point	θ_l	θ_h	Val F1	Test F1	Test AUC	Test MCC	Defer %	Eff Lat (ms)	Gap Closed
A	Budget-10	0.375	0.700	0.894	0.878	0.953	0.798	10.0	10.7	40%
A	Budget-20	0.150	0.700	0.910	0.905	0.964	0.842	20.5	13.6	64%
A	Balanced	0.025	0.925	0.948	0.935	0.976	0.892	46.8	20.9	92%
A	Latency-15	0.225	0.850	0.917	0.913	0.966	0.856	26.2	15.2	72%
A	Aggressive	0.025	0.975	0.952	0.940	0.978	0.900	51.8	22.3	96%
B	Budget-10	0.325	0.600	0.884	0.896	0.948	0.790	8.3	10.3	28%
B	Budget-20	0.125	0.600	0.922	0.915	0.957	0.829	16.1	12.4	43%
B	Balanced	0.050	0.900	0.967	0.965	0.976	0.931	36.5	18.0	84%
B	Latency-15	0.125	0.700	0.935	0.924	0.962	0.846	19.5	13.4	50%
B	Aggressive	0.050	0.975	0.976	0.974	0.981	0.948	45.5	20.5	91%
C	Budget-10	0.375	0.575	0.897	0.894	0.957	0.799	8.8	10.4	38%
C	Budget-20	0.275	0.675	0.925	0.923	0.969	0.855	19.5	13.4	66%
C	Balanced	0.250	0.875	0.944	0.941	0.977	0.890	31.3	16.6	82%
C	Latency-15	0.250	0.750	0.936	0.932	0.973	0.873	24.1	14.6	74%
C	Aggressive	0.025	0.975	0.958	0.961	0.990	0.926	66.0	26.1	101%
D	Budget-10	0.175	0.900	0.987	0.984	0.995	0.971	10.4	10.9	73%
D	Budget-20	0.050	0.925	0.990	0.989	0.996	0.980	17.4	12.8	88%
D	Balanced	0.200	0.750	0.985	0.980	0.994	0.964	6.5	9.8	62%
D	Latency-15	0.025	0.925	0.990	0.990	0.996	0.981	21.9	14.0	90%
D	Aggressive	0.025	0.975	0.991	0.992	0.997	0.986	25.9	15.1	98%

This is the cascade’s best case, and it depends on the low-cost arm being weak. CSC usually does not exceed the transformer arm on F1. The only exception is a marginal Dataset C gain, where the band tuner retains high-confidence low-cost predictions that the transformer would otherwise overturn. Its advantage is the F1–latency–interpretability trade-off it offers, which matters in IoT/WSN operations with different priorities: low-

cost gateway inference, SOC-side accuracy, or interpretability for warning banners. The advantage narrows as the low-cost arm strengthens, as Section 4.5 shows.

Across the four corpora at the Aggressive operating point, the hand-engineered CSC-HE-DistilBERT reaches a mean test F1 of 0.967 (Table 6), close to the frozen transformer arm at a deferral rate that keeps a substantial interpretable majority on the low-cost arm. On Dataset C, the Aggressive cascade (F1 = 0.961) marginally exceeds the DistilBERT-only baseline (0.959) because the band tuner can selectively retain high-confidence low-cost-arm predictions that the transformer arm would otherwise overturn.

4.3.2. Validation-Tuned Cascade (Held-Out Test Reporting)

To rule out the possibility that the cascade band was tuned on the test set, we re-run the cascade under a strict validation protocol. Band tuning happens after both arms are fixed, so it uses a validation slice carved from the held-out partition rather than the 60/20/20 split used to select the low-cost arm's configuration (Section 4.2.3); both stages keep selection off the final test data. The 20% held-out test partition is split 50/50 into a validation and a final test partition. The band is selected on the validation half only and evaluated once on the final test half. Table 6 reports the held-out test F1 of the CSC-HE-DistilBERT cascade at five validation-tuned operating points across all four corpora.

As a sensitivity check, re-selecting the band directly on the full test partition shifts each operating point by at most 0.01 F1 because the band-sweep landscape is broad. All cascade numbers reported in this paper use validation-only selection.

We also compare the CSC's probability-band policy against three simple alternatives at deferral budgets {10%, 20%, 40%}: *margin* (defer smallest $|p_L(x) - 0.5|$), *entropy* (defer highest binary entropy of $p_L(x)$), and *fixed-rate uncertainty*. These three alternatives are formally equivalent for a calibrated low-cost arm symmetric around 0.5, and the band, constrained to a contiguous probability interval, is marginally weaker at low budgets (mean test F1 across the four corpora 0.893, 0.919, and 0.950 at the 10%, 20%, and 40% budgets, against 0.916, 0.937, and 0.958 for margin and entropy). When fixed-budget F1 is the sole objective, the margin and entropy rules are the natural choice. The band rule is not F1-optimal. We keep it for operational auditability and monitoring: it is a single pair of scalars (θ_l, θ_h) on the calibrated probability axis, so each threshold maps to an interpretable confidence level, and the deferral rate is observable per email as a drift signal. Section 4.5.3 explores the complementary option of deferring on an explicit distribution-shift score rather than on uncertainty, which helps for monitoring but not for in-domain F1.

4.4. Cross-Corpus Generalisation

4.4.1. External Validation on Datasets B, C, and D

We replicate the same five pipelines on each of the three additional corpora. Table 7 gives the held-out test scores for the proposed pipeline across all four corpora. The proposed pipeline reaches a mean F1 of 0.878 (range 0.840–0.965), and the same qualitative ranking among feature sets (combined > custom-only > text-only) holds on every corpus.

Table 7. Held-out performance of the proposed pipeline on each of the four corpora (combined feature set + SMOTE + Bagging, identical 80/20 stratified split with `random_state = 42`). Source names are abbreviated Hugging Face identifiers; Dataset C is a stratified subsample of puyang2025/seven-phishing-email-datasets.

DS	Source	<i>n</i>	IR	F1	AUC	MCC
A	zefang-liu	18,634	1.55	0.840	0.943	0.746
B	jmmrcp/phishing-IBM (dedup)	11,505	1.02	0.851	0.929	0.704
C	puyang2025/seven-phishing	29,996	1.13	0.857	0.942	0.730
D	dorrabin/phishing_emails	30,651	1.22	0.965	0.992	0.936

4.4.2. Four-Dataset Cross-Transfer Matrix

To probe whether the pipeline survives distribution shifts at scale, we populate the full 4×4 train $\times$ test cross-corpus matrix, plus a union row in which the model is trained on A+B+C+D training partitions only (Table 8).

Table 8. The 4×4 cross-corpus F1 of the proposed pipeline; diagonal entries are in-domain references. The last row is the union model A+B+C+D evaluated on each corpus's held-out partition. In the header, rows are the training corpus ($\downarrow$) and columns the test corpus ($\rightarrow$).

Train $\downarrow$ Test $\rightarrow$	A	B	C	D
A	0.840	0.562	0.430	0.001
B	0.497	0.851	0.699	0.007
C	0.515	0.795	0.857	0.012
D	0.537	0.594	0.480	0.965
A+B+C+D	0.807	0.829	0.851	0.965

The mean off-diagonal F1 is 0.428 vs. a mean diagonal F1 of 0.878, a mean drop of 0.450. The worst direction is A $\rightarrow$ D (F1 = 0.001) because D's body distribution differs so sharply that A-trained features almost never fire. The union-trained model reaches $F1 \in \{0.807, 0.829, 0.851, 0.965\}$ on the four held-outs, within 0.04 F1 of each corpus's in-domain ceiling. The evaluation protocol transfers across corpora, but fitted single-corpus models do not. Practitioners deploying across heterogeneous mailflows should train on the union.

4.4.3. Per-Source Heterogeneity on Dataset C

Dataset C aggregates seven historically distinct phishing email corpora. We train two models, a within-C model and an external (A+B+D) model never seeing any C row, and evaluate both per-source on C's held-out partition (Table 9). The within-C F1 spans 0.55 (Ling-Spam, SpamAssassin) to 0.94 (CEAS-08), a 0.39-point range concealed by the single 0.857 Dataset-C F1. CEAS-08 is the hardest corpus for the external model ($\Delta F1 = +0.51$) and the easiest for the within-C model, indicating it is highly learnable in-domain and highly non-transferable out-of-domain. Practitioners deploying against a known mix of source corpora should report per-source metrics rather than the mean.

Table 9. Per-source F1 on Dataset C held-out. Within-C is trained on C's 80% training partition; external (A+B+D) is trained on the union of A+B+D training partitions and has not seen any C row.

Source	<i>n</i>	F1 (Within-C)	F1 (ext)	$\Delta F1$
CEAS-08	1141	0.935	0.426	+0.510
TREC-05	1689	0.834	0.439	+0.394
TREC-07	1543	0.888	0.611	+0.277
TREC-06	527	0.701	0.418	+0.283
Ling-Spam	78	0.550	0.389	+0.161
Enron	878	0.816	0.668	+0.148
SpamAssassin	144	0.563	0.541	+0.021

4.5. The Strong Classical Baseline and the Classical-First Cascade

The high-vocabulary classical baseline pairs a 10,000-feature word TF-IDF with a Platt-calibrated Linear SVM (Table 4). It reaches $F1 = 0.961$ on Dataset A. This matches the CPU-feasible fine-tuned DistilBERT reference (0.961, Section 4.2.4) and exceeds the frozen DistilBERT mean-pool baseline (0.950). The model costs 0.23 MB on disk and 0.22 s to fit. This raises a direct question: if a 0.23 MB classical model already matches the transformer,

is a low-cost-to-transformer cascade justified at all? We answer this in two steps. First, we subject the strong baseline to the same stress axes as the cascade. Then, we build the cascade on top of it.

4.5.1. Cross-Corpus Transfer of the Strong Classical Baseline

Table 10 reports the 4×4 train $\times$ test transfer matrix for the strong classical baseline. Its mean off-diagonal F1 is 0.734, against 0.428 for the hand-engineered low-cost arm (Table 8). The high-vocabulary classical model therefore transfers markedly better than the hand-engineered features. Single-corpus transfer is still uneven: the D-trained model drops to F1 0.26–0.47 off-diagonal. The union-trained model, in contrast, recovers $F1 \geq 0.95$ on every corpus.

Table 10. Cross-corpus F1 of the strong calibrated classical baseline (word TF-IDF 10k + Linear SVM). Diagonal entries are in-domain held-out references; the last row is the union model trained on A+B+C+D. Mean off-diagonal F1 = 0.734, against 0.428 for the hand-engineered arm (Table 8). In the header, rows are the training corpus ($\downarrow$) and columns the test corpus ($\rightarrow$).

Train $\downarrow$ Test $\rightarrow$	A	B	C	D
A	0.961	0.964	0.860	0.832
B	0.762	0.990	0.800	0.671
C	0.933	0.969	0.958	0.945
D	0.255	0.347	0.466	0.999
A+B+C+D	0.961	0.976	0.952	0.995

4.5.2. Adversarial and Calibration Robustness

The strong classical baseline is also more robust and better calibrated than the hand-engineered arm (Table 11). We apply the six attacker-side body transforms of Section 3.5.6. Its worst single attack is filler-padding ($\Delta F1 = -0.053$), and the combined attack costs only $\Delta F1 = -0.002$. The hand-engineered arm loses -0.183 under the same combined attack. The difference is that a 10,000-token vocabulary does not hinge on the few style and length cues the attack removes. Calibration tells the same story. After Platt scaling, the SVM reaches ECE 0.0145 and Brier 0.0233 against 0.0292 and 0.0937 for the hand-engineered arm. The uncalibrated SVM margin is poorly calibrated (ECE 0.187), so Platt scaling is both necessary and sufficient here.

Table 11. Deployment stress comparison of the hand-engineered low-cost arm and the strong calibrated classical baseline on Dataset A. Off-diagonal F1 is the mean of the cross-corpus matrices (Tables 8 and 10); $\Delta F1$ (combined) is the combined attacker-side transform (Sections 3.5.6 and 4.7.1); ECE/Brier are after Platt scaling; LLM-style F1 is from the stress set of Section 4.7.2.

Axis	Hand-Engineered (15-Feat)	Strong SVM (10k, calib.)
In-domain F1 (A)	0.847	0.961
Cross-corpus off-diag. F1	0.428	0.734
$\Delta F1$, combined attack	-0.183	-0.002
Calibration ECE (Platt)	0.0292	0.0145
Calibration Brier (Platt)	0.0937	0.0233
LLM-style F1	0.000	0.464
Disk (MB)	43	0.23
Latency (ms/email)	8.0	8.0

The one regime in which the strong classical baseline also fails is stylistically neutral, LLM-style phishing. On the synthetic stress set of Section 4.7.2, it reaches $F1 = 0.46$ (precision 0.57, recall 0.39), comparable to the frozen MiniLM (0.49) and DistilBERT (0.46) arms and

far above the hand-engineered arm ($F1 = 0$). No model we evaluate on this set, classical or transformer, exceeds $F1 \approx 0.49$.

4.5.3. Classical-First Cascade (CSC-SVM-DistilBERT) and Novelty-Gated Deferral

We next instantiate CSC with the strong calibrated classical model as the low-cost arm and frozen DistilBERT as the transformer arm (Table 12). Under the validation-tuned protocol of Section 4.3.2, CSC-SVM-DistilBERT reaches $F1 = 0.963$ on Dataset A. It defers 8.6% of messages and runs at 10.4 ms effective latency. For comparison, the SVM low-cost arm alone reaches 0.961 at 8.0 ms, and frozen DistilBERT alone reaches 0.950 at 35.5 ms. The cascade therefore beats the frozen transformer on both $F1$ and latency. It improves on the strong low-cost arm by only 0.002 $F1$. This gain is well within the paired-bootstrap variability of the headline comparison (Section 4.2.3) and should be read as a tie, not a ranked improvement. Once the low-cost arm is strong, the transformer arm has almost no in-domain headroom to recover. The cascade's value here is operational. It offers a single trade-off parameter, a low-cost-arm majority with inspectable lexical weights, and a monitorable deferral rate.

We also test deferral on a distribution-shift signal instead of calibrated uncertainty. Each message gets a low-cost nearest-neighbour novelty score: one minus its maximum cosine similarity to a training subsample in the TF-IDF space. The score separates the LLM-style distribution from the legacy test distribution with AUROC 0.70, a useful drift alarm. As a routing rule, it stays within 0.01 $F1$ of the probability band at matched budgets (Table 12), so we keep it for monitoring rather than routing.

Table 12. Classical-first cascade (CSC-SVM-DistilBERT: strong calibrated SVM low-cost arm, frozen DistilBERT transformer arm) on Dataset A, validation-tuned (Section 4.3.2). Lower block: Novelty-gated vs. probability-band deferral $F1$ at matched budgets, and the drift-detection AUROC of the novelty score for separating the LLM-style distribution from the legacy test distribution.

Configuration	F1	Defer %	Eff. Lat. (ms)
SVM low-cost arm only	0.961	0	8.0
Frozen DistilBERT only	0.950	—	35.5
CSC-SVM-DistilBERT (val-tuned)	0.963	8.6	10.4
<i>Novelty-gated/probability-band deferral F1 (in-domain)</i>			
Budget 10%	0.964/0.966	10	—
Budget 20%	0.964/0.961	20	—
Budget 40%	0.962/0.955	40	—
Novelty-score drift AUROC (LLM-style vs. legacy A test) = 0.70			

4.6. Calibration and Operational Thresholds

4.6.1. Calibration

Phishing classifiers are increasingly used downstream of a triage layer that consumes the *probability*, not the binary label. Table 13 reports Brier and ECE before and after Platt scaling.

Table 13. Calibration of the proposed compact pipeline. Lower Brier and ECE are better. Platt scaling is post hoc and does not affect AUC.

Pipeline	Brier	ECE	AUC	F1
Compact (raw)	0.0944	0.0345	0.940	0.840
Compact (Platt-calibrated)	0.0937	0.0292	0.941	0.843

The raw pipeline is already reasonably well calibrated ($ECE = 0.0345$). Platt scaling reduces ECE by 0.005 ($\approx 15\%$ relative) and Brier by 0.0007 with no loss in AUC, and the reliability curve sits close to the diagonal across the full probability range. Production deployments that consume the probability directly should apply Platt scaling; deployments that consume only the binary label may skip it. Calibration matters operationally because triage workflows consume the confidence score directly: under-confident phishing predictions waste analyst time, and over-confident legitimate predictions cause unnecessary user-facing warnings.

Table A1 sweeps the compact pipeline's decision threshold. A deployment intolerant of false positives (gateway warning-banner regime) can run at threshold 0.80 and tolerate 393 FPs per 10,000 legitimate emails at the cost of missing 33.5% of phishing; a recall-prioritising deployment (security-operation triage feed) can run at threshold 0.40 and surface 89% of phishing at 1479 FPs per 10,000 legitimate emails. The literature's two "95%-of-something" operating points give recall = 0.503 at threshold 0.92 and precision = 0.683 at threshold 0.155; the compact pipeline cannot simultaneously meet 95% precision and 95% recall on A.

4.6.2. Operational Threshold Across Six Models

For gateway deployment, the preferred operating point may not maximise F1. The metric of false positives per 10,000 legitimate emails is operationally interpretable because even a small false-positive rate can generate a large review burden at enterprise scale. Table A2 sweeps the threshold across six models on Dataset A: the fine-tuned DistilBERT reference attains the best recall at high-precision operating points, the frozen-arm cascades track their transformer arms closely, and the low-cost arm alone cannot reach the 95%-precision regime above a recall of 0.503.

4.7. Robustness and Error Analysis

4.7.1. Stylistic Stress-Test Boundary Analysis

Threat model. We model a moderate-budget attacker who can edit the body of their own phishing messages but cannot (i) poison the defender's training data, (ii) query the deployed classifier interactively, or (iii) read the model's gradients. Each transformation we apply corresponds to a low-cost real-world attacker strategy: `remove_exclamation` corresponds to avoiding spam-filter heuristics that key on excessive punctuation; `lengthen_filler` corresponds to padding messages with neutral business prose to dilute the suspicious-token density; `remove_keywords` corresponds to evading lexical-allow/deny filters by paraphrasing trigger terms; `unshorten_urls` corresponds to using full URLs rather than shorteners that risk-scoring services often flag; and `lowercase_all` corresponds to normalising the stylistic anomaly that uppercase-ratio features detect. The transformations are intentionally semantic-preserving from the attacker's point of view: the recipient still receives a click-bait message. We apply each transformation to the phishing class of A's held-out test set, re-extract features through the production pipeline, and re-score (Table 14).

The pattern matches the LOO importance ranking (Section 4.1.5): removing the dominant feature (`num_exclamation`) and inflating the length statistics (`lengthen_filler`) are the most damaging single attacks, each costing ≈ 0.13 F1 and ≈ 0.20 recall; removing keywords costs only 0.027 F1, and un-shortening URLs is essentially free for both parties. The *combined* attack drops F1 to 0.657, a 22% relative degradation, confirming that the low-cost arm remains vulnerable to low-cost style adaptation. These transformations are stress tests, not a complete adaptive adversarial evaluation, and they are directly relevant to phishing against IoT/WSN administrators: attackers can cheaply modify message style, length, urgency language, and URL presentation without changing the social-engineering objective.

Table 14. Stylistic stress-test boundary analysis of the compact pipeline under attacker-side feature manipulation. $\Delta F1$ and Δrecall computed against the unmodified baseline ($F1 = 0.840$, $\text{recall} = 0.863$). Transformations applied to *phishing test samples only*; this is not an adaptive evasion evaluation.

Transformation	F1	Recall	$\Delta F1$	Δrec	$\Delta \hat{p}$
(unmodified)	0.840	0.863	—	—	—
remove_exclamation	0.715	0.662	−0.125	−0.200	0.18
lengthen_filler	0.711	0.657	−0.129	−0.205	0.17
remove_keywords	0.813	0.816	−0.027	−0.047	0.04
unshorten_urls	0.840	0.862	−0.000	−0.001	0.00
lowercase_all	0.828	0.841	−0.012	−0.021	0.02
combined	0.657	0.583	−0.183	−0.280	0.24

4.7.2. LLM-Style Phishing Stress Set (Exploratory)

To probe the limitation that the four corpora may under-represent modern LLM-assisted phishing campaigns, we construct a small exploratory stress set of 500 phishing-style and 500 legitimate counterpart messages across ten common phishing scenarios (Microsoft account alert, Google Drive sharing, university password reset, payroll update, invoice/payment, delivery notification, bank security alert, conference invitation, research grant, HR document request). The phishing variants are written in a modern conversational tone with proper grammar and punctuation, deliberately under-using the stylistic markers the low-cost arm relies on; the legitimate variants describe the same scenarios in a normal organisational tone. The messages are assembled from fixed scenario templates with seeded slot-filling, not generated by a language model, so the set is deterministic and the generation script is included in the reproducibility package. This is an *exploratory probe*, not a primary benchmark (Table 15).

Table 15. Exploratory LLM-style phishing stress test. Each model is trained on Dataset A (full corpus) and evaluated on the 1000-message synthetic stress set. *Deferral %* is the share of stress emails routed to the transformer arm under the Aggressive cascade band $[0.03, 0.98]$. The low-cost pipeline falls to $F1 = 0$ because the stylistic markers it depends on are absent in LLM-style messages and the cascade defers all traffic to the transformer arm.

Model	F1	Prec.	Rec.	MCC	Defer %
Hand-engineered (15-feat)	0.000	0.000	0.000	−0.333	—
MiniLM + LR + SMOTE	0.489	0.629	0.400	0.176	—
DistilBERT + LR + SMOTE	0.461	0.689	0.346	0.219	—
CSC-HE-MiniLM Aggressive	0.489	0.629	0.400	0.176	100.0
CSC-HE-DistilBERT Aggressive	0.461	0.689	0.346	0.219	100.0

The stress test identifies a vulnerability regime: stylistically neutral, LLM-like phishing messages remove the cues used by the low-cost arm. Under the Aggressive band, CSC routes these messages to the transformer arm, but transformer-arm performance remains imperfect ($F1 \approx 0.46$ – 0.49 , $\text{recall} \approx 0.35$ – 0.40). On this distribution, the cascade reduces to the transformer arm, and the transformer arm is the load-bearing component. This is a diagnostic probe, not a robustness proof; the deferral rate can be monitored as a candidate drift signal, with empirical validation as a deployed drift detector left to future work.

4.7.3. Error Analysis

We extract four quadrants from Dataset A’s held-out test partition: 50 false positives from the CSC-HE-DistilBERT Aggressive cascade, 50 false negatives, 50 messages where deferral *fixed* a low-cost-arm error, and 50 messages that were deferred but still misclassified. Each is auto-categorised by keyword and structural rules into one of eleven categories

(Table A3). The dominant patterns are short bodies (very-short-body bucket: 20 FP, 21 still-wrong) and body-only phishing without a URL (33 FN, of which 22 are fixed by deferral).

A complementary qualitative inspection of the compact pipeline's own errors on the 3,727-sample held-out partition (279 false positives and 201 false negatives at threshold 0.5) reinforces the same picture. *False positives* cluster on two patterns: First, *legitimate marketing newsletters* that contain heavy uppercase formatting, multiple exclamation marks, and sweepstakes/promotion language (e.g., a Lifetime TV sweepstakes reminder triggered uppercase_ratio = 0.14, num_exclamation = 3, and the keyword indicator). Second, *long forwarded Enron business threads* with multiple "original message" delimiters and embedded URLs that pattern-match the link-presence features. Both classes are precisely the ones a stylistic feature classifier would be expected to confuse.

False negatives cluster on three patterns: (i) *noisy/obfuscated phishing* that has stripped exclamation marks, lowercased everything, and substituted character variants ("cialiis", "aiium") so that neither the keyword indicator nor the punctuation features fire; (ii) *polished low-keyword phishing* that reads like an ordinary marketing email; and (iii) *short non-English phishing* (the corpus is English-skewed but contains a small Portuguese/Spanish minority). The first pattern is the same one our adversarial analysis targets: it shows the threat model is realistic, not synthetic.

Adversarial flips (429 phishing samples drop below the threshold under the combined attack) are dominated by samples that originally fired *only* on length and exclamation features—i.e., short, exclamation-heavy messages whose discriminative signal disappears once the punctuation is stripped and the body is padded with neutral filler. Notably, none of the top-five highest-confidence flips fired the URL-shortener or HTML-form features, consistent with the LOO finding that those features carry a minimal independent signal.

This qualitative picture matches the quantitative results in Section 4.7.1: the model's mistakes and the model's vulnerabilities are co-located in the same feature subspace (style + length). Defenders should add header-level and URL-domain features (which our pipeline deliberately excludes for scope) to break this co-location.

4.8. Operational Relevance and Boundary Evaluation

Because no public IoT/WSN administrator-mail corpus is available, we use two bounded probes to assess operational relevance: a synthetic operator-targeted scenario set and an operations-themed subset of Dataset A (Table 16). The first is a synthetic set of 960 operator-targeted messages. They span twelve IoT/WSN operations scenarios: gateway firmware-update portals, MQTT-broker credential alerts, cloud IoT-hub dashboards, OT/SCADA VPN resets, sensor-fleet consoles, OTA signing-key requests, NOC ticketing, network-monitoring logins, ICS vendor advisories, smart-grid meter-data portals, gateway certificate renewals, and device-provisioning APIs. Each phishing message is paired with a legitimate counterpart and written in a clean modern operational tone. The second target is a subset of Dataset A's held-out test partition. We keep the 748 messages whose bodies mention IT-operations or infrastructure terms (server, VPN, gateway, firmware, dashboard, network, admin, portal, and similar). This subset approximates operations-themed emails that may appear in administrator or SOC mailboxes (252 phishing, 496 legitimate). No corpus is relabelled as WSN data. The operator scenarios are clearly synthetic, and the operations-themed subset is a keyword-filtered slice of existing emails.

Two findings follow. The first is positive. On the operations-themed subset, the strong calibrated classical model detects phishing at F1 = 0.986. This is above every other model and well above the hand-engineered arm (0.874). These results suggest that the recommended low-cost arm remains effective on an operations-themed slice of the evaluated corpus, but they do not establish performance on real IoT/WSN operational

mail streams. The second finding is a caution. On the synthetic operator scenarios, every model performs poorly, as on the LLM-style stress set. The messages are clean and carry few lexical phishing cues, so recall is low. The best F1 is 0.450, again from the strong SVM. On this distribution, deferring to the transformer arm lowers F1 rather than raising it. CSC-SVM-DistilBERT routes 85% of the scenarios to frozen DistilBERT and takes on its lower F1 (0.254 against 0.450 for the SVM alone) because the transformer arm is weaker than the low-cost arm on these novel messages. The implication for IoT/WSN operations matches the rest of this paper. A calibrated classical low-cost arm is an effective, low-cost detector for the operational mail stream as it exists today. Operator-targeted clean phishing is an open problem that neither a transformer nor a cascade currently solves. For that regime, the deferral signal is better used for monitoring than for automatic routing. This result also sets a deployment policy: under a novelty or distribution-shift alarm, CSC should not automatically defer to the frozen transformer unless that transformer has been validated on the target mailflow. Otherwise, the novelty signal should trigger analyst review, threshold retuning, or model retraining.

Table 16. IoT/WSN operation case study. *Operator scenarios*: 960 synthetic operator-targeted messages in clean modern tone. *Real ops subset*: 748 real operations-themed messages from Dataset A’s held-out partition. Models are trained on Dataset A; CSC-SVM-DistilBERT defers the low-cost arm’s uncertain band to frozen DistilBERT.

Model	Operator Scenarios		Real Ops Subset	
	F1	Recall	F1	Recall
Hand-engineered (15-feat)	0.000	0.000	0.874	0.921
Strong SVM (calibrated)	0.450	0.312	0.986	0.984
MiniLM + LR	0.041	0.021	0.941	0.956
DistilBERT + LR	0.254	0.146	0.951	0.968
CSC-SVM-DistilBERT (Aggressive)	0.254	0.146	0.978	0.976

5. Discussion

5.1. Accuracy–Interpretability Operating Points

Table 17 summarises the deployment-oriented choice among the models. The recommended default is the high-vocabulary calibrated SVM: it reaches F1 = 0.961 on Dataset A at about 8 ms and 0.23 MB, slightly exceeding the frozen DistilBERT baseline (0.950) at a fraction of its cost. The 15-feature hand-engineered model is the choice when per-feature auditability is required, at roughly 0.11 F1 below the SVM. CSC adds a deferral budget and drift monitoring on top of the SVM (Section 5.6).

Table 17. Deployment-oriented model comparison and recommendation on Dataset A held-out. Latency is per-email end-to-end CPU inference; disk is the serialised model size. The final column states when each model is the appropriate choice.

Model	F1	Latency (ms)	Disk	Interpretability	Recommended When
High-vocabulary SVM (recommended)	0.961	8.0	0.23 MB	sparse weights	default: high F1 at low disk and latency
15-feature hand-engineered	0.847	8.0	43 MB	per-feature	maximum per-feature interpretability
DistilBERT (frozen) + LogReg	0.950	35.5	265 MB	dense + explainer	not required in-domain
CSC-SVM-DistilBERT	0.963	10.4	—	mixed	deferral budget, escalation, or drift monitoring

Suspected LLM-style phishing drift: Do not rely on a body-only model; add header, URL, authentication, and attachment signals.

5.2. Interpreting the Feature-Representation Effect

The proximate explanation is the imbalance ratio: $IR = 1.55$ sits well below the regime in which resampling techniques typically dominate. SMOTE's documented gains over baseline learners are largest at $IR \geq 10$ in fraud-detection and rare-disease applications [23,24]; for $IR < 2$ on tree-ensemble classifiers, the literature already reports near-noise improvements. Our central finding *does not* generalise to severely imbalanced regimes. What we do claim to generalise is the methodological prescription: report imbalance-handling effects only after controlling for the expressiveness of the feature set, and report effect sizes alongside p -values so that readers can distinguish marginal gains from substantive ones.

5.3. Signal Content of the Custom Features

Phishing emailing is a *generative* problem from the attacker's side: the message must induce a click. Producing a click against a population that has been trained to distrust unsolicited mail requires (a) an authority impersonation, (b) an urgency cue, or (c) an actionable element (link, attachment, form). The ten features we engineer capture exactly these three dimensions at the lexical and structural level. Stylistic features such as exclamation count and uppercase ratio are easily measurable proxies for urgency cues; length statistics distinguish theatrical phishing from verbose business mail; and `has_phishing_keywords` captures an explicit authority-impersonation lexicon. The LOO analysis confirms that these proxies do most of the work.

This is consistent with the early phishing detection literature [18,19]. It is not consistent with the implicit assumption in much of the recent transformer literature that hand-engineered features are a legacy worth replacing.

5.4. Practical Deployment

For a mail gateway scanning approximately 10^6 messages per day, a per-email inference latency of 8.0 ms (compact pipeline) versus 18.5 ms (transformer pipeline) is the difference between a ≈ 2 h 13 min daily scan and a ≈ 5 h 09 min daily scan on a single 2-core CPU, both feasible, but the compact pipeline halves the CPU-time bill. Critically, the compact pipeline retrains end-to-end from raw text in 7.7 s on commodity hardware; the transformer baseline requires a 118.8 s encoder embedding pass whenever new training data arrive. Attackers pivot, defenders must redeploy, and a model that takes two minutes per retrain (versus eight seconds) will lag the threat by an extra cycle.

The compact pipeline's serialised form is 43 MB (Bagging with 200 estimators) and could be reduced to a single-digit MB by dropping `n_estimators` with a small F1 cost. The transformer encoder weights alone are 90 MB on disk.

5.5. Interpretability and Audit

Every prediction from the compact pipeline decomposes into the values of fifteen features and the contribution of each tree in the bagged ensemble. An incident-response analyst can be told *why* a particular email was flagged—for example, “the email contains the phrase *verify account*, links to a `bit.ly` URL, exhibits a 17% uppercase ratio, and matches two of the model's TF-IDF features.” The transformer baseline admits no such decomposition without an additional explainer (LIME, SHAP, integrated gradients), each of which is itself an approximation and adds latency.

5.6. Deployment Value of the Calibrated Selective Cascade

The CSC turns the low-cost-vs-transformer question into one parameter, the deferral rate. Its value depends on the low-cost arm. With a weak low-cost arm, it recovers most of the low-cost-to-deep F1 gap at lower latency. With the strong classical low-cost arm

(Section 4.5), it adds only 0.002 F1 over the arm alone at a small latency cost from the deferred fraction. Its role is then the explicit deferral budget and the monitoring signal, not an accuracy or latency gain, and a large majority of traffic stays on the low-cost arm. The deferral rate and the novelty score of Section 4.5.3 also serve as drift signals. Fuller validation as a deployed drift detector is left to future work.

This leads to a simple recommendation. If the deployment only needs maximum in-domain F1 and low latency, use the calibrated high-vocabulary SVM alone. Use CSC only when the deployment also needs an explicit deferral budget, drift monitoring, or escalation of uncertain messages. Table 17 summarises the choice by deployment need.

The low-cost arm functions as a fast triage filter, not a semantic phishing detector. It is inspectable under known corpus conditions, and its dominant style and length signals make drift monitoring and periodic recalibration necessary. The cascade's deferral mechanism is intended to mitigate this by routing low-confidence messages to the transformer arm. As the operator-scenario evaluation shows (Section 4.8), this helps only when the transformer arm is the stronger model on the deferred messages; on stylistically clean phishing, where the transformer is also weak, deferral does not recover accuracy.

5.7. Implications for Operational Security

In IoT/WSN deployments, CSC is best interpreted as a human-facing security control for the operational layer, deployed near mail gateways, SOC triage systems, or gateway-management workflows to reduce phishing-driven credential compromise affecting sensor-network administrators [8]. Its principal operational value is selective resource allocation: IoT/WSN management infrastructure is frequently hosted on low-cost edge servers, gateway-adjacent machines, or modest virtual instances rather than GPU-equipped data-centre nodes, and spending low-cost inference on the majority of straightforward messages while reserving transformer inference for the uncertain minority lets such infrastructure run an email security filter that would otherwise be too costly to apply per message.

This role is complementary rather than substitutive. CSC does not detect routing attacks, jamming, malicious packets, compromised nodes, or anomalous sensor readings; it should be deployed alongside established IoT/WSN protections such as secure routing, authentication, device hardening, network monitoring, firmware integrity checks, and packet-level intrusion detection [5,6]. Within a layered architecture, CSC closes the human-facing initial-access gap that cryptographic and routing-layer controls do not address.

Finally, calibration and threshold monitoring are necessary because operational email distributions drift. Administrator and operator inboxes receive a shifting mixture of vendor notifications, ticketing traffic, firmware-update alerts, and targeted social engineering; the calibrated probability that CSC routes on must be audited against this drift, and the decision threshold re-tuned to the deployment's tolerance for false positives on legitimate operational mail (Section 4.6.2).

5.8. Implications of the Cross-Dataset, Robustness, and Threshold Analyses

The four analyses establish three points about generalisation and robustness. First, the headline result holds across corpora: on Datasets B, C, and D, the proposed pipeline reaches an F1 of 0.851, 0.857, and 0.965, a mean of 0.878 across the four corpora. Second, a single model does not transfer between corpora—the mean off-diagonal F1 in the 4×4 matrix is 0.428—whereas joint training on A+B+C+D recovers in-domain performance to within 0.04 F1 on *every* corpus. Third, the low-cost arm is only partially robust to attacker-side feature manipulation: removing exclamation marks or padding messages with filler each costs ≈ 0.13 F1, and the combined attack costs 0.18 F1. The cascade does not guarantee robustness under distribution shift. In the evaluated legacy-style corpora, deferral helps

when the transformer arm is stronger than the low-cost arm on uncertain messages. Under stylistically clean or operator-targeted phishing, however, the transformer arm may also be weak; in that regime, the deferral or novelty signal should trigger analyst review, threshold retuning, or retraining rather than automatic reliance on the frozen transformer.

5.9. Limitations

The evaluation is confined to English email body text from four public corpora collected before 2025, so it may under-represent modern LLM-assisted phishing with cleaner grammar and conversational personalisation. The LLM-style stress set probes this gap rather than replacing a contemporary phishing-telemetry corpus. Because the detector reads the body only, it is not a substitute for gateway signals such as SPF, DKIM, DMARC, URL and sender reputation, attachment scanning, or sandboxing, and the error analysis (Section 4.7.3) confirms that several failure categories—missing-header, attachment-dependent, and obfuscated-domain cases—would be caught by header- or URL-aware pipelines. The English-specific lexical features (notably `has_phishing_keywords`) are not evaluated for multilingual transfer.

The low-cost arm's strongest cues are corpus-dependent style and length signals (Sections 4.1.3 and 4.7.1), so a deployment needs drift monitoring, periodic recalibration, and integration with existing gateway, URL, and reputation signals. The stylistic and LLM-style stress tests are diagnostic probes rather than a full adaptive-adversary evaluation, which we leave to future work. The transformer arm is evaluated as frozen throughout, with a single CPU-feasible fine-tuned DistilBERT reference on Dataset A, so that the fine-tune number is a lower bound on what a GPU-budget defender would obtain. We did not benchmark on ARM-class edge hardware or deployed IoT/WSN gateways; the reported CPU latency should therefore be interpreted as relative model cost, not as deployment-certified edge or IoT-gateway latency. Finally, although this paper is framed for IoT/WSN operational security, the corpora are public phishing email datasets rather than WSN traffic or organisation-specific administration mailboxes: the results support CSC as a candidate email security layer for IoT/WSN operations but do not establish performance on real operational mail streams from deployed environments, which remains the central evaluation for future work.

Two further caveats apply. First, the result that a calibrated classical model matches transformers holds for these in-domain English corpora; it is not a general claim, and a 2024+ telemetry corpus or a higher-budget fine-tuned transformer could change the ordering. Second, the IoT/WSN operator scenarios are synthetic rather than collected from a deployment, and the novelty score is only a moderate drift detector (AUROC 0.70). We therefore do not claim established performance on real IoT/WSN operational mail or on ARM/edge hardware; the IoT/WSN sections are a relevance and boundary analysis, not a deployment validation. Under a novelty alarm, the safe operational response is to route to analyst review or trigger retraining, not to defer automatically to the transformer arm, which the operator-scenario results show can be the weaker model on the shifted distribution.

5.10. Security, Privacy, and Ethical Considerations

This work proposes a defensive phishing detection mechanism. (1) The system is intended for defensive phishing detection inside organisational mail-flow infrastructure. (2) The benchmark uses public corpora only; no private mailbox data are collected or released. (3) False positives may disrupt legitimate business communication; defenders should monitor false-positive rate per 10,000 legitimate emails (Section 4.6.2) and adjust the threshold to deployment tolerance. (4) False negatives may expose users to credential theft;

the cascade specifically aims to reduce false negatives on uncertainty-boundary emails by deferring them. (5) Body-only classification is not a substitute for authentication (SPF, DKIM, DMARC), URL reputation, attachment scanning, sender reputation, or sandboxing. (6) Attackers may adapt their templates if they infer the model's features or deferral policy; defenders should refresh training data and audit feature distributions on a regular cadence. (7) Approximately 50–80% of cascade decisions remain decided by the low-cost arm. In the CSC-HE variant, these decisions can be decomposed into fifteen interpretable feature values; in the recommended CSC-SVM variant, they remain inspectable through sparse TF-IDF coefficients and calibrated scores, but not through the fifteen-feature explanation interface. The CSC should be deployed as one component in a layered email security pipeline rather than as a standalone phishing-defence mechanism.

6. Conclusions and Outlook

This paper studied resource-constrained phishing email detection for the human-facing operational layer around IoT and WSN deployments. Our central empirical finding concerns four public corpora. A low-cost Platt-calibrated high-vocabulary classical model (word TF-IDF with a Linear SVM) slightly exceeds the frozen DistilBERT reference and is statistically indistinguishable from the CPU-feasible fine-tuned reference (0.961 on Dataset A), at 0.23 MB on disk and about 8 ms per email. It also transfers across corpora better than the hand-engineered arm (mean off-diagonal F1 0.73 vs. 0.43), resists low-cost stylistic evasions (combined-attack $\Delta F1 \approx -0.00$ vs. -0.18), and stays well calibrated after Platt scaling (ECE 0.015). Transformer inference is therefore not a prerequisite for competitive accuracy on in-domain phishing. The Calibrated Selective Cascade (CSC) builds on this model. CSC is a deployment policy, not a new classifier. With the strong classical model as its low-cost sparse-linear arm, CSC gives the defender a single trade-off parameter, keeps the majority of decisions on an inspectable model, and yields a deferral-rate and novelty-rate signal for drift monitoring. On the evaluated corpora, its in-domain accuracy gain over the low-cost arm alone is marginal (≤ 0.002 F1). Its value is operational rather than a raw F1 gain. This study also confirms three secondary points: feature representation matters more than imbalance handling for the low-cost arm, single-corpus models transfer poorly while union training recovers in-domain performance, and per-source reporting is necessary for composite corpora.

Stylistically neutral, LLM-assisted phishing remains difficult for all the models we evaluate. On a synthetic stress set written in a clean modern tone, F1 falls to about 0.46 for the strong classical and transformer arms, and to 0 for the hand-engineered arm. A low-cost nearest-neighbour novelty score flags the shift only moderately (AUROC ≈ 0.70). Improving detection in this regime is more useful than optimising an in-domain accuracy that is already saturated. For IoT/WSN environments, phishing defence belongs to the operational-security perimeter, alongside the protocol-level controls that traditionally define it. Our case study on operator scenarios and a real operations-themed subset (Section 4.8) shows where a calibrated low-cost-arm cascade can sit, namely at a mail gateway or an SOC triage layer. It complements rather than replaces the controls that secure the network itself. Future work should evaluate adaptive phishing generation, multilingual and header-/URL-aware inputs, deployment on real operational mail streams, and live gateway deployment with streaming-latency and analyst-feedback metrics.

Funding: This research was funded by the Ongoing Research Funding program, grant number ORF-2026-1464, King Saud University, Riyadh, Saudi Arabia.

Institutional Review Board Statement: This study used only publicly available, de-identified email corpora and did not involve human subjects; institutional review board approval was therefore not required.

Informed Consent Statement: Not applicable.

Data Availability Statement: The analysis code, feature matrices, trained pipelines, evaluation and statistical test outputs, result tables, and figure-generation scripts that support the findings of this study are available from the corresponding author upon reasonable request. The four public corpora used in this study are those of refs. [17,41–43], obtained from their canonical Hugging Face Hub sources; third-party corpora are not redistributed where their license terms prohibit it.

Acknowledgments: The author thanks the maintainers of the zefang-liu/phishing-email-dataset corpus; the maintainers of the underlying Nazario, Enron, and SpamAssassin archives; and the open-source authors of scikit-learn, imbalanced-learn, XGBoost, LightGBM, and sentence-transformers. Generative AI tools were used solely for language and formatting refinement; the author reviewed and approved all content and takes full responsibility for the work.

Conflicts of Interest: The author declares no conflicts of interest. The funder had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Appendix A. Additional Operating Point and Error Tables

Table A1 gives the full decision-threshold sweep for the compact pipeline (Section 4.6.1), Table A2 the high-precision/high-recall operating points across six models (Section 4.6.2), and Table A3 the error-quadrant breakdown for the CSC-HE-DistilBERT Aggressive cascade (Section 4.7.3).

Table A1. Operating points for the compact pipeline on A held-out. The held-out partition contains 2265 legitimate emails and 1462 phishing emails.

Thr.	Prec.	Rec.	F1	FP/10k Legit	FN/10k Phish
0.30	0.756	0.917	0.829	1907	834
0.40	0.795	0.891	0.840	1479	1094
0.45	0.804	0.873	0.837	1377	1265
0.50	0.819	0.863	0.840	1232	1375
0.55	0.830	0.852	0.841	1126	1484
0.60	0.843	0.828	0.835	998	1724
0.70	0.869	0.776	0.820	755	2237
0.80	0.916	0.665	0.771	393	3352
0.90	0.942	0.534	0.682	212	4658

Table A2. Operational threshold analysis across six models on Dataset A held-out (2265 legitimate, 1462 phishing). *Recall@95%P* is the highest recall achievable at precision $\geq 95\%$; analogously for 99%P and 95%R.

Model	R@95%P	R@99%P	P@95%R
Hand-engineered compact	0.503	—	0.683
MiniLM + LR + SMOTE	0.835	0.713	0.903
DistilBERT mean-pool + LR + SMOTE	0.884	0.754	0.942
Fine-tuned DistilBERT (CPU config)	0.973	0.863	0.956
CSC-HE-MiniLM Aggressive	0.857	—	0.910
CSC-HE-DistilBERT Aggressive	0.884	—	0.940

Table A3. Error-analysis quadrants on the CSC-HE-DistilBERT Aggressive cascade (Dataset A held-out). *FP/FN* = false positives/negatives; *Fixed* = low-cost arm wrong, cascade right; *Still wrong* = deferred but still misclassified.

Category	FP	FN	Fixed	Still Wrong
Newsletter/marketing	11	6	5	8
Very short body	20	6	9	21
Phishing with no URL	0	33	22	7
Attachment-dependent clue	3	1	4	0
Phishing formal business tone	2	1	1	1
Long thread with many links	2	0	0	1
Legit “security” language	0	1	1	0
Obfuscated brand/domain	1	2	1	1
Other/uncategorised	17	12	18	15

References

- Verizon Business. 2024 *Data Breach Investigations Report*; Verizon Business: New York, NY, USA, 2024.
- IBM Security. *X-Force Threat Intelligence Index 2024*; IBM Corp.: Armonk, NY, USA, 2024.
- Tang, M.; Lau, V.K.N. Online identification and temperature tracking control for furnace system with a single slab and a single heater over the wirelessly connected IoT controller. *IEEE Internet Things J.* **2024**, *11*, 6730–6747.
- Wang, B.; Ma, H.; Wang, F.; Dampage, U.; Al-Dhaifallah, M.; Ali, Z.M.; Mohamed, M.A. An IoT-enabled stochastic operation management framework for smart grids. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 1025–1034.
- Hassija, V.; Chamola, V.; Saxena, V.; Jain, D.; Goyal, P.; Sikdar, B. A survey on IoT security: Application areas, security threats, and solution architectures. *IEEE Access* **2019**, *7*, 82721–82743.
- Butun, I.; Österberg, P.; Song, H. Security of the Internet of Things: Vulnerabilities, attacks, and countermeasures. *IEEE Commun. Surveys Tuts.* **2020**, *22*, 616–644.
- Alkhalil, Z.; Hewage, C.; Nawaf, L.; Khan, I. Phishing attacks: A recent comprehensive study and a new anatomy. *Front. Comput. Sci.* **2021**, *3*, 563060.
- Xiao, Y.; Jia, Y.; Liu, C.; Cheng, X.; Yu, J.; Lv, W. Edge computing security: State of the art and challenges. *Proc. IEEE* **2019**, *107*, 1608–1631.
- Salloum, S.; Gaber, T.; Vadera, S.; Shaalan, K. Phishing email detection using natural language processing techniques: A literature survey. *Procedia Comput. Sci.* **2021**, *189*, 19–28.
- Tida, V.V.R.; Hsu, S. Universal spam detection using transfer learning of BERT model. *arXiv* **2022**, arXiv:2202.03480.
- Fang, Y.; Zhang, C.; Huang, C.; Liu, L.; Yang, Y. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. *IEEE Access* **2019**, *7*, 56329–56340.
- Ribeiro, M.T.; Singh, S.; Guestrin, C. ‘Why should I trust you?’: Explaining the predictions of any classifier. *arXiv* **2016**, arXiv:1602.04938.
- Nazario, J. Phishing Corpus. 2004–Present. Available online: <https://monkey.org/~jose/phishing/> (accessed on 1 July 2026).
- Klimt, B.; Yang, Y. Introducing the Enron Corpus. 2004. Available online: <https://www.ceas.cc/papers-2004/168.pdf> (accessed on 1 July 2026).
- Apache Software Foundation. *SpamAssassin Public Corpus*. 2002. Available online: <https://spamassassin.apache.org/old/publiccorpus/> (accessed on 1 July 2026).
- Aassal, A.E.; Baki, S.; Das, A.; Verma, R. An in-depth benchmarking and evaluation of phishing detection research for security needs. *IEEE Access* **2020**, *8*, 22170–22192.
- Liu, Z. *Phishing-Email-Dataset*. Hugging Face. 2023. Available online: <https://huggingface.co/datasets/zefang-liu/phishing-email-dataset> (accessed on 1 July 2026).
- Fette, I.; Sadeh, N.; Tomasic, A. Learning to detect phishing emails. In *Proceedings of the 16th International Conference on World Wide Web*; ACM: New York, NY, USA, 2007; pp. 649–656.
- Toolan, R.M.; Carthy, J. Feature selection for spam and phishing detection. In *2010 eCrime Researchers Summit*; IEEE: Piscataway, NJ, USA, 2010.
- Hamid, I.R.A.; Abawajy, J. Hybrid feature selection for phishing email detection. In *11th International Conference, Algorithms and Architectures for Parallel Processing*; Springer: Berlin/Heidelberg, Germany, 2011.
- Bagui, S.; Nandi, D.; Bagui, S.; White, R.J. Classifying phishing email using machine learning and deep learning. In *2019 International Conference on Cyber Security and Protection of Digital Services (Cyber Security)*; IEEE: Piscataway, NJ, USA, 2019.

22. Bountakas, P.; Koutroumpouchos, K.; Xenakis, C. A comparison of natural language processing and machine learning methods for phishing email detection. In *Proceedings of the 16th International Conference on Availability, Reliability and Security*; ACM: New York, NY, USA, 2023.
23. He, H.; Garcia, E.A. Learning from imbalanced data. *IEEE Trans. Knowl. Data Eng.* **2009**, *21*, 1263–1284.
24. Krawczyk, B. Learning from imbalanced data: Open challenges and future directions. *Prog. AI* **2016**, *5*, 221–232.
25. Galar, M.; Fernández, A.; Barrenechea, E.; Bustince, H.; Herrera, F. A review on ensembles for the class imbalance problem. *IEEE Trans. Syst. Man Cybern. C* **2012**, *42*, 463–484.
26. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **2002**, *16*, 321–357.
27. Han, H.; Wang, W.-Y.; Mao, B.-H. Borderline-SMOTE: A new over-sampling method in imbalanced data sets learning. In *Advances in Intelligent Computing, International Conference on Intelligent Computing, ICIC 2005*; Springer: Berlin/Heidelberg, Germany, 2005.
28. He, H.; Bai, Y.; Garcia, E.A.; Li, S. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*; IEEE: Piscataway, NJ, USA, 2008.
29. Elkan, C. The foundations of cost-sensitive learning. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*; ACM: New York, NY, USA, 2001.
30. Drummond, C.; Holte, R.C. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the ICML Workshop Learning from Imbalanced Datasets*, Washington, DC, USA, 21 August 2003.
31. Lemaître, G.; Nogueira, F.; Aridas, C.K. Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *J. Mach. Learn. Res.* **2017**, *18*, 1–5.
32. Alladi, T.; Chamola, V.; Sikdar, B.; Choo, K.-K.R. Consumer IoT: Security vulnerability case studies and solutions. *IEEE Consum. Electron. Mag.* **2020**, *9*, 17–25.
33. Chow, C.K. On optimum recognition error and reject tradeoff. *IEEE Trans. Inf. Theory* **1970**, *16*, 41–46.
34. Geifman, Y.; El-Yaniv, R. Selective classification for deep neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*; ACM: New York, NY, USA, 2017.
35. Viola, P.; Jones, M. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001*; IEEE: Piscataway, NJ, USA, 2001.
36. Teerapittayanon, S.; McDanel, B.; Kung, H.T. BranchyNet: Fast inference via early exiting from deep neural networks. In *2016 23rd International Conference on Pattern Recognition (ICPR)*; IEEE: Piscataway, NJ, USA, 2016.
37. Chen, L.; Zaharia, M.; Zou, J. FrugalGPT: How to use large language models while reducing cost and improving performance. *arXiv* **2024**, arXiv:2305.05176.
38. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140.
39. Jones, K.S. A statistical interpretation of term specificity and its application in retrieval. *J. Doc.* **1972**, *28*, 11–21.
40. APWG. *Phishing Activity Trends Report*. Quarterly. Anti-Phishing Working Group: Lexington, MA, USA. Available online: <https://apwg.org/trendsreports/> (accessed on 1 July 2026).
41. Reyes, J.M. *Phishing-IBM*. Hugging Face Hub. 2023. Available online: <https://huggingface.co/datasets/jmmrcp/phishing-IBM> (accessed on 1 July 2026).
42. Puyang. *Seven-Phishing-Email-Datasets*. Hugging Face Hub. 2025. Available online: <https://huggingface.co/datasets/puyang2025/seven-phishing-email-datasets> (accessed on 1 July 2026).
43. Rabin, D. *Phishing_emails-data*. Hugging Face Hub. 2024. Available online: https://huggingface.co/datasets/dorrabin/phishing_emails-data (accessed on 1 July 2026).
44. Batista, G.E.A.P.A.; Prati, R.C.; Monard, M.C. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explor.* **2004**, *6*, 20–29.
45. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: New York, NY, USA, 2016; pp. 785–794.
46. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree. In *Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 4–9 December 2017.
47. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. *arXiv* **2019**, arXiv:1908.10084.
48. Sanh, V.; Debut, L.; Chaumond, J.; Wolf, T. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv* **2019**, arXiv:1910.01108.
49. Brier, G.W. Verification of forecasts expressed in terms of probability. *Mon. Weather Rev.* **1950**, *78*, 1–3.
50. Naeini, M.P.; Cooper, G.; Hauskrecht, M. Obtaining well calibrated probabilities using Bayesian binning. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI Press: Palo Alto, CA, USA, 2015.

51. Platt, J. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*; MIT Press: Cambridge, MA, USA, 1999; pp. 61–74.
52. Matthews, B.W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim. Biophys. Acta* **1975**, *405*, 442–451.
53. Dietterich, T.G. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Comput.* **1998**, *10*, 1895–1923.
54. Demšar, J. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.

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.