E-PhishGen: Unlocking Novel Research in Phishing Email Detection

Luca Pajola*
SpritzMatter S. R. L.
Padua, Italy
University of Padua
Padua, Italy
luca.pajola@spritzmatter.com

Eugenio Caripoti*
SpritzMatter S. R. L.
Padua, Italy
University of Padua
Padua, Italy
eugenio.caripoti@spritzmatter.com

Stefan Banzer*
University of Liechtenstein
Vaduz, Liechtenstein
stefan.banzer@uni.li

Simeone Pizzi*
SpritzMatter S.R.L.
Padua, Italy
simeone.pizzi@spritzmatter.com

Mauro Conti
University of Padua
Padua, Italy
Orebro University
Orebro, Sweden
mauro.conti@unipd.it

Giovanni Apruzzese
University of Liechtenstein
Vaduz, Liechtenstein
University of Reykjavik
Reykjavik, Iceland
giovanni.apruzzese@uni.li

Abstract

Every day, our inboxes are flooded with unsolicited emails, ranging between annoying spam to more subtle phishing scams. Unfortunately, despite abundant prior efforts proposing solutions achieving near-perfect accuracy, the reality is that countering malicious emails still remains an unsolved dilemma.

This "open problem" paper carries out a critical assessment of scientific works in the context of phishing email detection. First, we focus on the *benchmark datasets* that have been used to assess the methods proposed in research. We find that most prior work relied on datasets containing emails that—we argue—are not representative of current trends, and mostly encompass the English language. Based on this finding, we then re-implement and re-assess a variety of *detection methods reliant on machine learning* (ML), including large-language models (LLM), and release all of our codebase—an (unfortunately) uncommon practice in related research. We show that most such methods achieve near-perfect performance when trained and tested on the same dataset—a result which intrinsically hinders development (how can future research outperform methods that are already near perfect?). To foster the creation of "more challenging benchmarks" that reflect current phishing trends, we propose E-PhishGEN, an LLM-based (and privacy-savvy) framework to generate novel phishing-email datasets. We use our E-PhishGEN to create E-PhishLLM, a novel phishing-email detection dataset containing 16616 emails in three languages. We use E-PhishLLM to test the detectors we considered, showing a much lower performance than that achieved on existing benchmarks—indicating a larger room for improvement. We also validate the quality of E-PhishLLM with a user study (n=30). To sum up, we show that phishing email detection is still an open problem—and provide the means to tackle such a problem by future research.

CCS Concepts

• **Security and privacy** → **Phishing**; • **Computing methodologies** → **Machine learning**.

Keywords

benchmark, dataset, large language models, email, spam, detection

ACM Reference Format:
Luca Pajola, Eugenio Caripoti, Stefan Banzer, Simeone Pizzi, Mauro Conti, and Giovanni Apruzzese. 2025. E-PhishGen: Unlocking Novel Research in Phishing Email Detection. In *Proceedings of the 2025 Workshop on Artificial Intelligence and Security (AISec '25), October 13–17, 2025, Taipei, Taiwan.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3733799.3762967

1 Introduction

Hardly a day goes by without hearing of yet another company being targeted by a phishing attack [10, 19]. In particular, phishing *emails* still represent one of the most common vectors to penetrate an organization and carry out any sort of cyberattack—ranging from gathering login credentials, installing malware, or stealing industrial secrets [10]. Simply put, the reality is that the battle against phishing-email attacks is never in favor of system defenders.

And yet, by turning the attention at prior research on *phishing email detection*, the conclusions of a large number of papers seem to align: phishing emails can be detected with near-perfect performance by using diverse techniques within the machine-learning (ML) domain. For instance, Bountakas and Xenakis [28] claim a random forest classifier achieves 98.6% accuracy; Doshi et al. [36] achieve 98.4% accuracy with a logistic regression model; whereas Atawneh and Aljehani [21] use a BERT model obtaining 99.6% accuracy; large-language models (LLM) have also been used, achieving accuracy above 95% [25, 64]. Put simply, the scenario portrayed in

*Luca Pajola is the first author and should be contacted for correspondence about this work. Eugenio Caripoti and Stefan Banzer have both contributed substantially to this work. SpritzMatter S.R.L. is a spinoff of the University of Padua.

AISec '25, Taipei, Taiwan

ACM ISBN 979-8-4007-1895-3/2025/10
https://doi.org/10.1145/3733799.3762967

research depicts a remarkably different, and almost contradictory, reality than what is faced by real-world companies.

In this work, we seek to *(i)* provide evidence that the detection of phishing emails is an open problem *in research*, *(ii)* identify potential root causes, and *(iii)* devise the means to address such a problem by future research. Let us explain how we pursue our goals.

Research Questions and Findings. We begin our quest with Section §2, which revolves around a broad, and our first, research question (RQ1): *"what benchmark datasets are used in related literature to assess previously-proposed phishing email detectors?"* To answer RQ1, we review related literature, and find that most prior works relied on a subset of eight datasets, which share three problems: *(a)* they mostly include English emails—neglecting other languages, such as German or Italian, in which phishing emails can be written; *(b)* they often mix "phishing" with "spam" emails—the latter not necessarily posing a security/privacy risk; *(c)* they predominantly include emails collected before 2010—which are unlikely to reflect current phishing trends. To provide a clear case, we report in Email 1 a sample contained in the SpamAssassin dataset [39] (used, e.g., to test recent approaches proposed in [21, 36, 64]). Altogether, these findings show the limited scope of existing benchmark datasets, i.e., a crucial component in an experimental evaluation.

[Subject] Help!

You have been specially selected to qualify for the following:
Premium Vacation Package and Pentium PC Giveaway
To review the details, please click on the link below using the confirmation number:
http://www.1chn.net/wintrip
Confirmation Number: **Lh340**
Please confirm your entry within 24 hours of receiving this confirmation.
Wishing you a fun-filled vacation!
If you have any additional questions or cannot connect to the site, do not hesitate to contact me:

vacation@btamail.net.cn

Email 1. An email in the popular dataset SpamAssassin [39] (from 2005).

Then, in Section 3, we tackle our second research question (RQ2): "*what performance do existing detectors achieve on some previously-used benchmark datasets?*" Indeed, we found that few prior works released their codebase, and even though the results of prior works aligned, it was difficult to pinpoint a clear baseline. Hence, to facilitate future research, we carry out a reproducible and statistically-validated *reassessment of previously-proposed phishing email detectors* on the identified benchmark datasets. We consider detection techniques reliant on feature-engineering (e.g., TF-IDF) as well as those based on "feature-agnostic" techniques such as BERT, and we also consider detectors that leverage pretrained large-language models (LLMs) in a zero-shot fashion. Our results confirm that ML methods trained and tested on the same dataset achieve near-perfect performance—which is not a very encouraging result, because it indicates that existing benchmarks are not adequate to measure the "improvement" that a given method may have over existing ones. As a potential workaround, we also carry out a cross-evaluation [18] by testing our considered detection methods on different datasets. Our results show an (expected) drop in performance—suggesting that a similar (but not very common, in this domain) practice may be more appropriate to gauge the quality of novel detectors.

Next, in Section 4, we turn the attention to our third research question (RQ3): "*what is a way to overcome the shortcomings of existing datasets—without raising privacy concerns?*" Indeed even by mixing existing datasets, their limited scope (e.g., few languages, old emails) does not enable one to assess the real-world effectiveness of existing detectors. A potential solution is collecting emails from "our inboxes", but such a practice may raise privacy concerns [47], especially given that a benchmark dataset must be publicly released. Therefore, we *(a)* propose a framework, E-PhishGEN, that enables researchers to create custom datasets of phishing emails that conform to user-provided characteristics; and *(b)* use our proposed framework to create E-PhishLLM, a novel dataset of 16616 emails (equally split between benign and phishing) encompassing different languages and "personal profiles", thereby overcoming the shortcomings of existing datasets.

Finally, in Section 5, we focus on our fourth and fifth research questions, specifically: "*what performance do previous methods achieve on E-PhishLLM?*" (RQ4) and "*does E-PhishLLM contain phishing emails of a higher quality than those included in previously-proposed datasets?*" (RQ5). Answering these RQs serves to *(i)* provide a well-founded baseline for future work that seeks to use E-PhishLLM for benchmarking purposes, but also *(ii)* validate our claim that E-PhishLLM overcomes commonly used benchmark datasets. For RQ4, we test the methods considered in RQ2 on the new dataset, whereas for RQ5 we carry out a user study with experts (n=30).

We discuss limitations and ethics (including the potential "dual use" of E-PhishGEN) in Section 6. We release all of our resources [11].

Technical Contributions. Despite being primarily an "open problem" paper, we do make three technical contributions to the domain of applied ML for cybersecurity. Namely:

- We propose E-PhishGEN, an original framework to generate targeted phishing emails tailored for user-specific profiles.
- We provide E-PhishLLM, a novel dataset of phishing emails—generated via E-PhishGEN—that are qualitatively better than those contained in existing datasets (validated via an user study).
- We reassess previously-proposed and ML-based phishing email detection methods on existing datasets as well as on E-PhishLLM.

In summary, we provide the tools to "unlock" the relatively stagnant (in our opinion) state of research on phishing email detection.

2 Literature Review and Motivation

We briefly introduce the problem of phishing email detection (§2.1), then we address RQ1 (§2.2), and finally discuss related work (§2.3). Importantly, we focus on phishing *emails*: other phishing vectors (e.g., websites [12], SMS [62], voice [42]) are outside our scope.

2.1 Background: Phishing Emails

Phishing is a form of cyberattack that has been plaguing internet users by over twenty years [23, 35]. The overarching principle of phishing is to leverage social-engineering techniques that induce a given victim to give the attacker what they want. For instance, an employee may receive an email from their supervisor (whose real email address has been compromised or spoofed) asking to open an attachment (which may contain, e.g., ransomware); or they may receive a fake email from their security department asking to change their login credentials, pointing to a malicious link [63].

Phishing Emails (in the real world). Every year, new reports from renown sources (e.g., Proofpoint [8], APWG [19], Cisco Talos [9], or even the FBI [7]) reveal that companies of all sizes are constantly, and increasingly, targeted by phishing threats—especially

after the recent release of LLMs [8, 63]. Among the most common recommendations against such a (almost never-ending) problem is phishing training/education [48, 49, 63]. However, such defensive practices are still not widespread [8], and even "trained" employees may occasionally fall victim to a phishing scam, potentially compromising their entire organization [49, 63]. Therefore, there is a need to develop automated detection techniques that can prevent end-users from receiving (or opening) a phishing email in the first place. In practice, this is typically done via hard-coded mechanisms such as blocking/flagging emails from unknown senders, or that contain attachments with a known malicious signature, or that include specific terms/URLs in the text [26, 40, 63]. However, all such methods can be easily circumvented by real-world attackers (after all, we do receive a lot of phishing emails!) denoting that currently-deployed detectors/filters present huge margins for improvement.

Phishing Email Detection (in research). A variety of approaches, typically reliant on machine learning (ML), have been proposed in research to mitigate the problem of phishing emails. At a high-level, such methods seek to overcome the intrinsic limitation of "static" filters, which can only work against phishing emails that match pre-defined rules (e.g., inclusion of an URL reported in a blocklist). Development of ML-based detectors typically requires three steps [17]: First, a representative dataset of phishing and benign emails must be collected; potentially, the dataset must be pre-processed to enable the application of ML methods. Second, a given ML model must be trained on the (pre-processed) dataset. Third, the ML model must be tested (on different data): if the performance of such ML model meets certain requirements (e.g., low false positives with high true positives) then it can be deployed in operational environments. Broadly, we can distinguish three categories of ML-based phishing-email detectors:

- *Feature-based.* These detectors analyse manually-engineered features (e.g., occurrence of certain words in the email's text). Therefore, such methods require a feature-extraction process that turns the original email into a feature-vector, representing the sample provided as input to the ML model. For instance, Doshi et al. [36] apply TF-IDF on the email's text before sending the sample to the ML-based classifier; their evaluation revealed that the best-performing model relied on the logistic regression (LR) algorithm.
- *Feature-agnostic.* These detectors send the whole email directly to the ML model—and hence do not require any pre-processing step. However, it is necessary to carry out some training/fine-tuning of the model. For instance, Fang et al. [38] use a convolutional neural network that receives the output of the Word2Vec algorithm to perform their detection; whereas Lee et al. [50] (as well as [21]) use language models, such as BERT.
- *LLM-based.* These detectors can work without any preprocessing or fine-tuning. The intuition is to leverage "large" language models to autonomously carry out the detection task, potentially by issuing one/few prompts—as done in, e.g., [25, 64].

Of course, a detector can also rely on a combination of the aforementioned categories, e.g., by combining different detectors in an ensemble, pipeline, or high-level architecture (as done, e.g., in [50]).

2.2 Datasets for Phishing Email Detection [RQ1]

In research, whenever a paper proposes any given solution, such a paper should demonstrate that the proposed method "outperforms the state of the art" (or "achieves state of the art performance"). Such a demonstration requires an evaluation—which necessitates a dataset. As we wrote, the evaluations of previously-proposed methods for phishing email detection typically show near-perfect performance—a finding confirmed by recent literature reviews [14, 29, 33, 47, 59, 61]. *If prior research shows such a stunning performance, then why is it that phishing emails keep flooding our inboxes?* Such a dilemma led us to scrutinize the datasets used to test the methods of prior work—i.e., RQ1. Let us explain how we "motivate" our paper.

Methodology. We acknowledge that prior reviews have tackled a similar question (e.g., [14, 29, 33, 47, 59, 61]); moreover, investigating all literature on phishing-email detection is unfeasible. To provide a meaningful, and humanly-possible, answer to RQ1 we carry out a semi-systematic literature review [66], proceeding as follows. **(1)** We search well-known repositories (Google Scholar, IEEE Xplore, Tailor&Francis) with queries "[LLM/large language model/] phishing email [detection/classification]", and we integrate our results with papers accepted to top-conferences (WWW, S&P/EuroS&P, CCS/AsiaCCS, NDSS, USENIX SEC, ACSAC, IMC, WDSM, CHI) in 2014–2024 that have "phish" in the title (as also done in [63]). This led to 562 papers (for Google Scholar, we only considered the first 100 results returned with each query). **(2)** We filtered our papers by looking at their metadata. For instance, we used the abstract to determine if the paper was truly about "phishing email detection" (and not, e.g., phishing "website" detection), and we generally excluded unpublished articles. This led us to a set of 100 papers. **(3)** We qualitatively analysed the full-text of these papers. Specifically, attention was put on the following aspects: *(i)* what datasets are used? *(ii)* what detection methods are proposed? *(iii)* is the source code available? Such a qualitative analysis has been carried out by four researchers, who worked independently and discussed their findings in meetings (as also done in [15]).

Qualitative Findings. We report the results of our dataset-centered analysis (detection approaches are covered in §3).

- *Overly-used datasets.* The vast majority of our analysed papers relies on the same datasets to test their methods: SpamAssassin [39], Enron [45], SpamBase [43], Nazario [54], LingSpam [58], NigerianFraud/Clair [56], TREC [32], CEAS [3]. While reliance on the same dataset is not a bad practice per-se (after all, it enables comparisons across different methods), it becomes such when one considers that such datasets (aside from Nazario) have emails collected mostly before 2010. Such "old" data clearly does not reflect the most recent phishing trends.[1] Moreover, such datasets are monolingual (i.e., they only include English text), preventing one from gauging the effectiveness of any given method against phishing emails in different languages. Finally, such datasets have labeling issues because oftentimes spam emails are labeled as phishing—despite being semantically very different.
- *Lack of source code.* Only few papers release their source code (e.g., [13]); in some cases, we found a link to empty repositories (e.g., [51]). This is problematic because, in various cases (e.g., [36, 51]), the dataset itself is generated by mixing different datasets (sometimes even "enriched" with synthetic emails [25],

[1] We argue that 99% accuracy on detecting phishing emails exchanged *in 2005* cannot be used to claim that a method is an "efficient detector of phishing emails" after 2020.

or collected from the authors' inboxes [28]. Hence, lack of open-source code not only prevents replicability of the method, but also impairs reproduction of the same testbed by future research.
- *No clear naming.* We found that prior works may refer to a certain dataset with different names. For instance, SpamBase is referred to as "UCI machine learning repository" in [21] and "UCI repository" in [13]; whereas the experiments in [30] are carried out on a subset of the Enron dataset—which is, however, referred to as "the publicly available Phishing Email Detection dataset [10]", and reference [10] (in [30]) links to a dataset on Kaggle for which no source was specified, and only a comment from an user highlighted that the samples are taken from Enron. More generally, many papers (see [47]) refer to datasets from Kaggle (instead of the true source), preventing accurate attribution.

Due to the last points, *we cannot provide a precise number* of the occurrences of any given dataset in prior work. Indeed, names are often mixed, and lack of source code prevents verifying which dataset was actually used to test a method.

Less popular datasets. There exist other datasets/repositories usable to test phishing-email detectors. Among these, we mention: Untroubled [2] (used in [24]); PhishingPot [6] (used in [34]); PhishBowl [5] (used in [55]); IWSPA [4] (used in [65]); MillerSmiles [1] (used in [46]); as well as the recent dataset by Chataut et al. [31]. Such "less-used datasets" may not be affected by the issues of the "overly-used datasets". However, we report that none of the papers we analysed that considered such datasets released their code; moreover, some datasets are "dead" (e.g., IWSPA has less than a dozen emails at the time of writing this article) or require a payment (e.g., only a portion of MillerSmiles is free). Finally, and importantly, some of these datasets (e.g., PhishingPot) are not designed to be "benchmarks", but rather are community-driven efforts to collect phishing emails, and hence they change continuously. Without source code, it is impossible to exactly replicate the same testbed.

Answer to RQ1. Previously proposed methods are evaluated on datasets (such as SpamAssassin, SpamBase, Enron, Nazario, LingSpam) that have old (and monolingual) emails—which hardly resemble current phishing trends. Moreover, related literature often does not release their codebase: this is problematic especially given that it prevents accurate replication of the testbed.

2.3 Related Work (and Novelty)

To avoid misunderstandings and clarify our focus, let us position our paper within extant literature on phishing email detection.

Thousands of papers have tackled the problem of phishing emails. Various reviews/systematizations have analysed (or re-examined) the performance of diverse ML-based detectors, as well as the datasets used to test such detectors (see, e.g., [14, 29, 33, 47, 59, 61]). We acknowledge these contributions, which is why we do not claim novelty in our aforementioned analysis (§2.2). Indeed, our analysis serves as a scaffold to highlight that—in the phishing-email detection context—there is an "open problem" (i.e., the constant re-use of not-very-representative datasets) that deserves to be broadcast.

We acknowledge that our re-assessment (discussed in §3), entail previously-proposed detection methods. Again, we do not claim novelty in such an evaluation. Indeed, by releasing all of our resources, our goal is to provide a solid foundation that can be used to spearhead future research in the phishing-email detection domain.

In contrast, we do claim novelty in our proposed E-PhishGEN framework (§4). Even though methods to generate synthetic dataset exists (e.g., [52]), we are not aware of any prior work that proposed a methodology that automatically *(i)* generates "profiles" of potential targets of phishing emails, and *(ii)* crafts high-quality phishing emails tailored to such profiles. We also claim originality in our E-PhishLLM dataset (especially given that it is multilingual), whose quality has been validated with an user study (§5).

3 Reassessment of Previous Detection Methods

In our literature analysis, we found: a shortage of publicly-available source code on phishing email detection; as well as the lack of a "standardized" testbed (due to the mixing of various datasets, which may have labelling issues [13, 14]). Hence, to provide a foundation for future work, we carry out a comprehensive reassessment of previously proposed ML-based detection methods.

In doing so, however, we go beyond the traditional evaluation methodology of testing a model on data from the same distribution as that of the training set (done, e.g., in [14]): we will also evaluate the generalizability of the considered detectors by applying cross-evaluation methodologies [18]. We first present our setup (§3.1), then present the results (§3.2), and then draw considerations (§3.3).

3.1 Experimental Setup

Recall RQ2: "what performance do existing detectors achieve on some previously-used benchmark datasets?" To provide a meaningful—but humanly-feasible—answer to RQ2, we carry out our reassessment by considering eight detectors evaluated across eight datasets.

Datasets (and standardization) We use eight publicly available email datasets, spanning across popular (e.g., Enron) and less-used ones (e.g., Chataut [31]). Table 1 provides an overview of their sizes, class distributions, and sources. Importantly, to provide an evaluation that aligns with prior work, we considered "variants" of these datasets used by some works for which we found a dedicated repository (i.e., [13, 29, 41]). Note that some datasets (i.e., Enron and LingSpam) are listed twice: once for the variant by [13] and the second for [41]. We expect our results on these datasets to be similar (given that they are drawn from the same distribution), hence such a design choice can be used to validate our results. To enable cross-evaluations, it is necessary that all datasets are in the same format [18]. To this end, we standardize our considered datasets by clearly separating *subject*, *bodytext*, and *label* of each email. We also cleaned the text by, e.g., removing multiple whitespaces, or HTML tags (given that not all datasets include them).

Detection Methods We explore eight techniques pertaining to the three categories of ML-based detectors mentioned in §2.1. Specifically: five feature-based classifiers—Logistic Regression (LR), Naive Bayes (NB), Random Forest (RF), Support Vector Machine (SVM), and MultiLayerPerceptron (MLP)—which rely on TF-IDF representations for the email subject and bodytext (similarly to [27, 36, 59]); one feature-agnostic and transformer-based model—DistilBert (DB)—which will undergo a fine-tuning process (similarly to [44]); and two state-of-the-art LLMs—gemini-2.0-flash and gpt-4o-mini—with zero-shot prompting (as also done in [57]).

Evaluation Protocol The common practice is to test an ML model on data drawn from the same dataset used to train/fine-tune such an ML model: such a protocol prevents one from gauging the effects of combining data from different distributions. Hence, to "maximize" the potential of existing datasets, we explore the additional scenarios enabled by cross-evaluations [18]. Specifically, we design our evaluation around three core experiments.

- *Experiment-1*: we scrutinize whether models requiring a training/fine-tuning phase—which typically achieve strong performance when tested on samples from the same dataset—suffer from a degraded performance when tested on samples from different datasets. This is useful to measure the generalization capabilities.
- *Experiment-2*: we examine an "all-vs-one" scenario (inspired by [16]) in which we train/fine-tune a model on data from 7 (out of 8) datasets, and we then test such a model on the left-out dataset. This is useful to see if the combination of existing datasets can cover some blind-spots, leading to models which better generalize—potentially at the expense of a lower performance on data from the same training distribution. (To our knowledge, this experiment is new in the phishing-email detection context.)
- *Experiment-3*: we gauge how well LLMs can act as phishing-email detectors in a zero-shot context (the prompt is in our repo [11]).

To provide statistically significant results, we repeat our experiments five times—each time by applying a stratified 70:30 train-test split, but with a different random seed. The performance is always measured on the "test" partition (to avoid data leak [20]). Due to space limitations, we will report only the average F1-score (useful to combine both the true- and false-positive rate) in this paper: the complete results (e.g., accuracy, precision, recall, as well as standard deviations) are in our repository [11].

Dataset Name	Size	# Phishing	# Benign	Variant
CEAS [3]	39126	21829	17297	[13]
Enron-v1 [45]	29569	13778	15791	[13]
Ling-v1 [58]	2797	445	2352	[13]
SpamAssassin [39]	5791	1704	4087	[13]
TREC [32]	123232	55291	67941	[29]
Chataut [31]	24583	19681	4902	[31]
Enron-v2 [45]	9601	4687	4914	[41]
Ling-v2 [58]	2590	423	2167	[41]

Table 1: Overview of the datasets used in our reassessment.

3.2 Results [RQ2]

Due to space limitations, we will report only the average F1-score (useful to combine both the true- and false-positive rate) in this paper: the complete results (e.g., accuracy, precision, recall, as well as standard deviations) are in our repository [11]

Experiment-1. We report the results in Table 2. All our considered models exhibit poor cross-dataset generalization. For instance, LR and NB achieve very high F1-scores when tested on data from the same training distribution (e.g., 0.98 and 0.83 on CEAS, respectively) but their performance drops drastically in an inter-dataset context (down to 0.27 and 0.01, with average drops of 0.51 and 0.69, respectively). DistilBERT, while more robust, follows a similar trend: the performance on the same dataset is near-perfect (e.g., 1.00 on CEAS) but the performance drops on different datasets (e.g., 0.67 on SpamAssassin); yet, DistilBert has a much smaller average performance drop (e.g., fine-tuned on TREC, the average drop is

Model	Trained On	CEAS	Enron-v1	Ling-v1	SpamAssassin	TREC	Chatuat	Enron-v2	Ling-v2	Average Drop
Logistic Regression	CEAS	0.98	0.57	0.29	0.32	0.68	0.57	0.58	0.27	0.51
	Enron-v1	0.74	0.96	0.43	0.51	0.77	0.83	0.96	0.44	0.30
	Ling-v1	0.45	0.62	0.91	0.73	0.54	0.35	0.62	0.92	0.31
	SpamAssassin	0.42	0.65	0.74	0.91	0.55	0.40	0.66	0.71	0.32
	TREC	0.83	0.92	0.68	0.73	0.94	0.59	0.92	0.65	0.18
	Chatuat	0.72	0.63	0.25	0.45	0.62	1.00	0.65	0.26	0.49
	Enron-v2	0.72	0.95	0.41	0.47	0.65	0.87	0.95	0.42	0.31
	Ling-v2	0.49	0.65	0.93	0.73	0.56	0.39	0.66	0.93	0.30
Naive Bayes	CEAS	0.83	0.14	0.01	0.05	0.25	0.35	0.15	0.01	0.69
	Enron-v1	0.79	0.95	0.60	0.58	0.78	0.75	0.96	0.62	0.23
	Ling-v1	0.70	0.71	0.97	0.54	0.67	0.71	0.72	0.96	0.25
	SpamAssassin	0.67	0.70	0.62	0.95	0.68	0.45	0.71	0.63	0.31
	TREC	0.79	0.90	0.83	0.79	0.88	0.51	0.90	0.82	0.08
	Chatuat	0.69	0.62	0.27	0.45	0.60	0.98	0.64	0.28	0.48
	Enron-v2	0.79	0.96	0.59	0.58	0.78	0.75	0.96	0.62	0.23
	Ling-v2	0.70	0.71	0.97	0.54	0.68	0.71	0.73	0.96	0.24
Random Forest	CEAS	0.99	0.52	0.23	0.12	0.58	0.51	0.53	0.21	0.60
	Enron-v1	0.73	0.98	0.44	0.54	0.80	0.78	0.99	0.46	0.30
	Ling-v1	0.47	0.72	0.98	0.71	0.59	0.42	0.72	0.99	0.32
	SpamAssassin	0.46	0.69	0.68	0.97	0.63	0.40	0.70	0.67	0.36
	TREC	0.84	0.94	0.58	0.77	0.97	0.54	0.94	0.57	0.23
	Chatuat	0.72	0.64	0.28	0.45	0.62	1.00	0.66	0.28	0.48
	Enron-v2	0.71	0.97	0.41	0.52	0.79	0.80	0.97	0.43	0.31
	Ling-v2	0.50	0.72	0.99	0.71	0.60	0.44	0.73	0.98	0.31
Support Vector Machine	CEAS	0.99	0.61	0.28	0.30	0.72	0.55	0.62	0.28	0.50
	Enron-v1	0.77	0.97	0.44	0.51	0.78	0.83	0.97	0.45	0.29
	Ling-v1	0.42	0.70	0.96	0.77	0.60	0.38	0.70	0.94	0.32
	SpamAssassin	0.47	0.68	0.72	0.94	0.59	0.41	0.69	0.68	0.34
	TREC	0.84	0.94	0.70	0.74	0.95	0.60	0.94	0.67	0.18
	Chatuat	0.72	0.64	0.27	0.45	0.62	1.00	0.66	0.28	0.48
	Enron-v2	0.73	0.97	0.40	0.48	0.72	0.87	0.96	0.41	0.31
	Ling-v2	0.44	0.70	0.98	0.77	0.62	0.40	0.71	0.97	0.31
Multi-Layer Perceptron	CEAS	0.99	0.69	0.40	0.46	0.77	0.54	0.69	0.41	0.43
	Enron-v1	0.76	0.98	0.60	0.60	0.83	0.74	0.98	0.61	0.25
	Ling-v1	0.54	0.66	0.97	0.79	0.61	0.42	0.67	0.97	0.30
	SpamAssassin	0.64	0.70	0.75	0.97	0.68	0.43	0.71	0.73	0.31
	TREC	0.82	0.95	0.66	0.74	0.97	0.57	0.95	0.65	0.21
	Chatuat	0.72	0.64	0.28	0.45	0.62	1.00	0.66	0.28	0.48
	Enron-v2	0.73	0.98	0.62	0.42	0.65	0.75	0.97	0.64	0.29
	Ling-v2	0.53	0.66	0.98	0.79	0.61	0.41	0.67	0.96	0.30
DistilBERT	CEAS	1.00	0.83	0.62	0.67	0.83	0.56	0.84	0.57	0.30
	Enron-v1	0.80	0.99	0.58	0.60	0.88	0.77	1.00	0.55	0.26
	Ling-v1	0.81	0.71	1.00	0.52	0.69	0.75	0.72	1.00	0.25
	SpamAssassin	0.84	0.81	0.74	0.98	0.80	0.56	0.81	0.77	0.22
	TREC	0.86	0.98	0.89	0.84	0.99	0.56	0.98	0.87	0.14
	Chatuat	0.71	0.64	0.28	0.45	0.62	1.00	0.66	0.28	0.48
	Enron-v2	0.83	0.99	0.52	0.56	0.84	0.80	0.99	0.49	0.27
	Ling-v2	0.81	0.69	1.00	0.52	0.68	0.74	0.69	0.98	0.25

Table 2: Cross-evaluation results (Experiment-1). We report the (averaged over 5 trials) F1-scores achieved by each model when trained on each dataset (rows) and tested on any dataset (columns). The last column shows the average drop in F1-score when testing on other datasets compared to the diagonal.

0.14—albeit the F1-score is still only 0.56 on Chatuat). Such a result indicates that embedding-based models (such as DistilBERT) might have better generalization capabilities w.r.t. those that rely on vocabularies built via TF-IDF. Finally, we appreciate that all models achieve high performance when tested on data from the same "variant" of a given dataset (e.g., the SVM trained on Enron-v1 has 0.97 F1-score on Enron-v1, and 0.97 F1-score on Enrong v2; and viceversa): this (expected) result validates our experimental setup.

Experiment-2. We report the results of the models trained on 7 datasets and tested on the left-out dataset in Fig. 1. Perhaps surprisingly, the MLP (which uses TF-IDF) has the most consistent performance (from 0.89 on TREC, to 0.98 on Chatuat) across all datasets: in comparison, DistilBERT has much wider margins (from

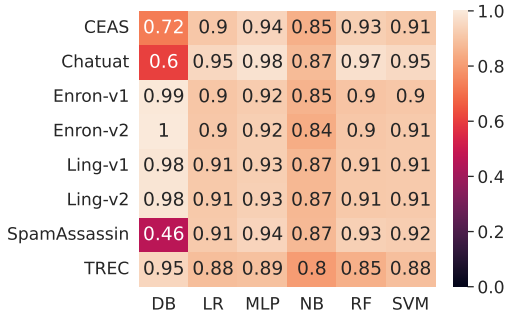


Fig. 1: All-vs-one results (Experiment-2). We train each model (columns) on seven datasets, and test it on the left-out dataset (rows).

0.46 on SpamAssassin, to 1.00 on Enron-v2). This indicates that, if there is availability of large amounts of data from different distributions, it may be wiser to use ML models reliant on TF-IDF (such as the simple MLP or RF) rather than embedding-based models (such as DistilBert). Finally, to gauge if an expanded training set leads to any performance degradation on data from the same distribution, we measure the (averaged) F1-score of the models trained on each of these all-but-one combinations of datasets and compare it with the (averaged) F1-score of the "diagonals" in Table 2. The results are in Table 3. We can see a substantial drop in the same-dataset performance—which is an expected result. Indeed, this is the cost of developing ML models with better generalizability.

Model	*Single*	*AllvsOne*
LR	0.95	0.90
NB	0.94	0.85
MLP	0.98	0.92
RF	0.98	0.91
SVM	0.97	0.91
DB	0.99	0.95

Table 3: Performance on data from the same training set. We report the average F1-score achieved by the models in Experiment-2 (AllvsOne column) and those in Experiment-1 (Single column) when tested on data from the same training set.

Experiment-3. We report the results of the LLMs in Figure 2. Both Gemini-2.0-Flash and gpt-4o-mini achieve F1-scores above 0.86, with the only exception being on the Chataut dataset. Intriguingly, even DistilBERT (which also uses transformers) struggled in the all-vs-one scenario on the Chataut dataset (see Fig. 1). This result may indicate that the Chataut dataset is either substantially different from the others, or that there may be some labelling issues (e.g., benign emails labeled as phishing, or vice-versa).

Answer to RQ2. We confirm that models requiring a training phase achieve near-perfect performance when tested on data from the same dataset. Yet, such models struggle when tested on data from other datasets—but DistilBERT seems to have a better generalization power. Zero-shot-prompted LLMs exhibit high F1-scores when tasked to detect phishing emails from our considered datasets, but the performance on Chataut is poor.

3.3 Considerations

Let us discuss and position our findings within extant work.

First, our reassessment includes five repeated trials for each experiment, and each trial entails having all methods tested on the same test portion of each dataset. Such a procedure therefore enables one to carry out *statistical analyses* to gauge if any given

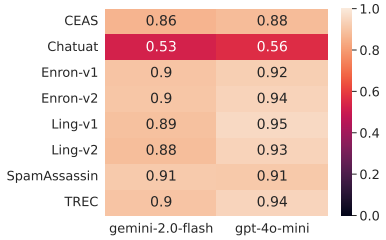


Fig. 2: LLM performance (Experiment-3).

method is better than another. For instance, in the same-dataset setting (i.e., the diagonal in Table 2) one would find that the two models with highest average F1-score across all datasets are the MLP 0.977 (avg: 0.977, std: 0.016) and the RF (avg: 0.978, std: 0.011); these results are produced by 40 evaluations (5 trials on 8 datasets): a t-test reveals that these two methods can be considered as statistically equivalent (p<.05). Notably, such statistical tests are not mentioned even in popular reviews (e.g., [14, 29, 33, 47, 59, 61]). Our repository [11] includes all the data used to compute such tests.

Second, we showed that cross-evaluations are a good way to test the generalizability of prior methods, since models with high performance on the same dataset exhibit a substantial drop when tested on a different dataset. Yet, there is a problem: mixing different datasets may create *temporal bias* (see [20]) because the samples in the training set may come from datasets collected after those used in the test set (e.g., Chataut was released in 2024, whereas SpamAssassin in 2005). Moreover, existing datasets do not reflect current phishing trends (e.g., they do not have LLM-generated emails, which are popular today [8, 63]) and are mostly monolingual.

The aforementioned problems led us to RQ3 ("*what is a way to overcome the shortcomings of existing datasets—without raising privacy concerns?*"), the answer of which is in the next section.

4 Proposed E-PhishGEN Framework

We now present our proposed E-PhishGEN (short for "Email Phishing Generator") framework—the answer to RQ3. In principle, we want to provide a tool that enables researchers to *(i)* automatically generate a large corpus of emails to develop/test phishing-email detectors, which *(ii)* are of high-quality and can reflect current trends, such as being LLM-written, while *(iii)* not raising privacy concerns related to using emails from personal inboxes.

We first describe our E-PhishGEN framework (§4.1 to §4.3), and then use it to generate our proposed E-PhishLLM dataset (§4.4).

4.1 Overview

E-PhishGEN is designed as a modular pipeline for generating realistic, diverse, and context-aware synthetic emails—including both benign (ham) and malicious (phishing) examples. It leverages LLMs to simulate plausible organizational contexts and communications, enabling the creation of high-quality datasets for phishing detection research. The framework is composed of two primary modules:

(1) *Profile Generation.* This module uses an LLM to generate synthetic company and user profiles from minimal input prompts or high-level descriptors (e.g., industry sector, company size, or regional context). Given these inputs, the LLM constructs realistic corporate structures and employee personas, including names, roles, departments, and hobbies. These profiles serve

as the foundation for creating personalized and coherent email communications in subsequent steps.

(2) *Email Generation.* Building on the profiles produced in the previous module, this module employs the LLM to craft emails that reflect realistic communications. It produces both legitimate (ham) messages—such as meeting requests, announcements, or transactional updates—and phishing emails tailored to the recipient's role and organizational context. Phishing variants cover a range of attack vectors, including credential harvesting, malicious attachments, and social-engineering scams.

The combination of a role-aware context and varied phishing strategies allows the generation of challenging and diverse training data.

4.2 Module 1: Profile Generation

This module generates realistic organizational contexts based on a user-defined country. The process unfolds in two steps:

(1) *Company Generation.* Given a selected *Country* (e.g. United States, Italy), the framework generates X synthetic companies that reflect the local economic and cultural context. This step influences the language and tone of emails produced in later stages. Companies are described using meaningful attributes such as sector, size, and region. Prompt 1 shows a simplified version of the prompt for generating the companies based on a country defined by the user

(2) *Employee Generation.* For each company, the system creates Y synthetic employee profiles that emulate realistic corporate diversity. Each profile includes key attributes, such as role, job title, seniority, and active projects, that condition the email generation process. The Prompt 2 shows a simplified version of the prompt to generate the user profile.

Note that *country*, X, Y are parameters of our framework.

Prompt 1: Company Generation (*Country, X*)

Generate X fictional companies based in $Country$, reflecting a realistic distribution in regional diversity, income, and sector. Each company should have these characteristics: Name, founding year, product / services, background history, headquarter location, number of employees, annual revenue, main clients, extent of the business.

Prompt 2: Profile Generation (*CompanyProfile, Y*)

Generate Y realistic employee profiles for a company, specifically create profiles of those who are likely to use a personal computer in their daily work. Ensure that a mixed but realistic set of employees is generated that covers senior, mid-level, and junior roles.
Company profile: {$CompanyProfile$}
Each user profile must include: Name, Gender, Age, Birthplace, Education, Languages, Role, Current Projects, Time employed in the company, Tech Proficiency, Hobbies, Social Media.

4.3 Module 2: Email generation

This module generates realistic emails—both benign (ham) and phishing—based on the synthetic company and employee profiles from Module 1. The generation process includes two main steps:

- *Email Scenario Generation.* Using the company and employee profiles as input, the framework produces N email scenarios, capturing key contextual attributes (e.g., topic, urgency, tone) that shape the content and intent of the email.
- *Email Content Generation.* Given the company, employee, and associated scenario, the LLM generates N full email texts tailored to each context.

Therefore, as X is a parameter, the framework will produce $X \times Y \times N$ emails for each country. In this module, the email scenario and content is highly dependent on the focus on the email, and we designed one for each of the ham and phishing categories scenarios. For instance, as shown in Prompt 3, one of the setting can be *type* that can be a legitimate or phishing email. Furthermore, the $EmailTraits$ defines the type of characteristics we need to generate the email. For the legitimate scenario: content description, sender, tone, style, length, and receiver info. For the phishing scenario: phishing type, customization level, objective, impersonated identity, method, social engineering technique, tone and style, length.

Prompt 3: Scenario Generation (*type, N, CompanyProfile, UserProfile, EmailTraits*)

Generate the following characteristics for N $type$ emails that the employee might receive based on $CompanyProfile$ and $UserProfiles$. Each email profile must contain: $EmailTraits$.

4.4 Proposed E-PhishLLM Dataset

We now demonstrate the application of our E-PhishGEN framework, and use it to generate our proposed E-PhishLLM dataset. We set ourself the objective to generate emails in three languages: English, Italian, and German.

For each language, we generate a series of company profiles for the relevant country (for English, we use both the UK and the US). Then, for each company, we generated 5 profiles of likely employees. Finally, for each employee, we generated 10 emails, five legitimate and 5 phishing. In total, we generated the following companies, spanning from the various countries: 250 between the UK and the US, 60 in Italy, and 50 in Germany. Note that as E-PhishGEN involves the utilization of LLM, some email generation procedures were erroneous and have been discarded (e.g., we expected a JSON as an output, and the LLM answer was not properly formatted). We utilized GPT-4o-mini from OpenAI as an LLM.

The E-PhishLLM dataset consists of a total of 16,616 emails. The class distribution is balanced, with emails equally distributed between legitimate and phishing. In terms of language, the majority of emails are in English (69.22%), followed by Italian (16.26%) and German (14.14%). This multilingual composition highlights the utility of E-PhishLLM for language-agnostic phishing detection strategies.

We now show an output example for an Italian company.


```
{"company_name": "Fabbri Tech Automazione", "
    ↪ establishment_year": "2003", "
    ↪ offered_products_services": "Industrial automation
    ↪ systems and robotics", "company_details": "Fabbri
    ↪ Tech reflects the growing trend towards automation
    ↪ in Italy's manufacturing sector, providing cutting-
    ↪ edge robotics solutions. Known for custom-tailored
    ↪ software and engineering services, it serves a
    ↪ variety of industries, establishing a strong
    ↪ reputation in the European market.", "
    ↪ headquarters_location": "Modena, Emilia-Romagna", "
    ↪ number_of_employees": "320", "annual_revenue": "
    ↪ EUR60 million", "main_consumer": "Manufacturers
    ↪ across various sectors including automotive and
    ↪ food processing", "affairs_extent": "International,
    ↪  with clients primarily in Europe and Asia."}
```

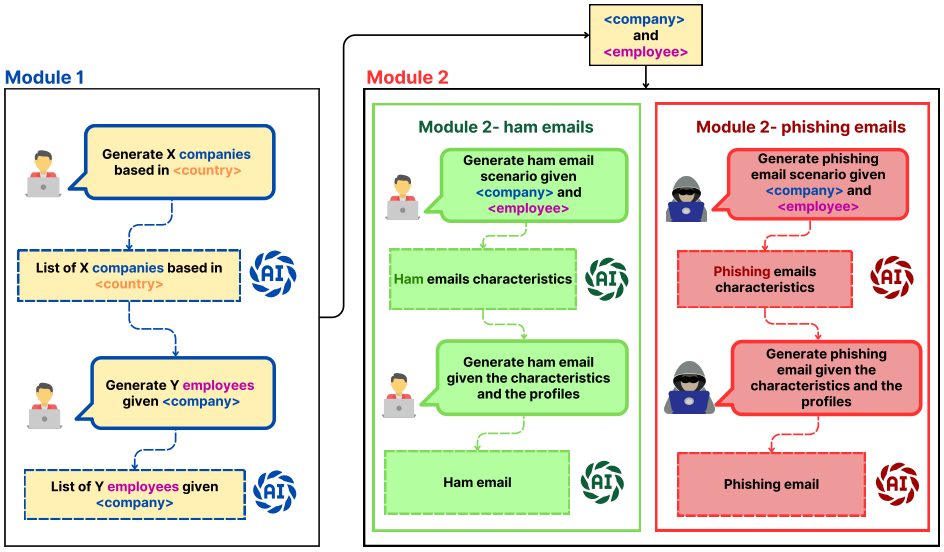


Fig. 3: Overview of E-PhishGEN.. The framework is composed of two modules: Module 1 generates synthetic company and employee profiles based on high-level input prompts; Module 2 takes these profiles as input and uses an LLM to generate realistic emails, including both benign and phishing variants, tailored to the organizational context and user roles.

The model generates a 'Tech Automatation' company based in 'Emilia-Romagna' (a well-known region famous for this type of company), with an annual revenue of "€60 million". The LLM uses this type of information to generate a likely employee.

```
{"name": "Marco Bianchi", "gender": "M", "age": 29, "
    ↪ birthplace": "Florence, Italy", "qualifications": "
    ↪ Bachelor's in Electromechanical Engineering,
    ↪ Universita degli Studi di Firenze", "languages": "
    ↪ Italian (Native), English (Intermediate), Spanish (
    ↪ Basic)", "job_title": "Junior Project Coordinator",
    ↪  "current_project": "Assisting in the coordination
    ↪ and planning of projects involving automated
    ↪ systems in automotive manufacturing", "
    ↪ time_employed": "2 years", "tech_proficiency": "
    ↪ Intermediate", "hobbies": "Traveling, Video gaming"
    ↪ , "social_media": "LinkedIn, Facebook"}
```

The employee, Marco Bianchi, has a bachelor's in Electromechanical Engineering, and his profile fits what is described by the company. Finally, with both information, we see two example of emails generated, one benign and one malicious.

[Subject] Scheduling a Call for Supply Chain Adjustments

Dear Marco, I hope this message finds you well. We need to schedule a video call to discuss some adjustments and potential delays in the supply chain affecting our current project components. Could you please inform me of your availability this week? Looking forward to hearing from you. Best regards, TomJohnson

Email 2. Illustrative example of a benign email in E-PhishLLM.

[Subject] Urgente: Verifica delle Credenziali dell'Account

Ciao Marco, Ti scrivo per conto del tuo manager per richiedere un'urgenteverifica delle tue credenziali aziendali. È molto importante che tuproceda al controllo immediato della correttezza delle informazioni d'accesso personali. Si prega di seguire il link di verifica di seguito e aggiornare qualsiasi informazione necessaria quanto prima:«link» Grazieperla tua collaborazione. Cordiali saluti, Federica Rossi Responsabile IT Fabbri Tech Automazione

Email 3. Illustrative example of a malicious email in E-PhishLLM.

By analyzing Marcos' email, we noticed that as the company operates in international markets, incoming emails are written in both Italian and English. We believe that this automation provided by LLM, provides a realistic scenario of working experience.

5 Benchmarking and Validation of E-PhishLLM

We now scrutinise our proposed E-PhishLLM dataset, answering RQ4 ("*what performance do previous methods achieve on E-PhishLLM?*") in §5.1; and RQ5 ("*does E-PhishLLM contain phishing emails of a higher quality than those included in previously-proposed datasets?*") in §5.2.

5.1 Performance Assessment [RQ4]

We test the detectors considered in §3 on our E-PhishLLM dataset. Importantly, we only consider the English portion of E-PhishLLM (encompassing 11502 emails in total, 5996 phishing and 5506 benign ones). This is to enable a fair comparison with our previously evaluated methods, which are tailored for English texts.

Method. We follow the same evaluation protocol as in §3.1, but the major difference is that, here, we use E-PhishLLM only to *test* existing methods. This is to follow the guidelines of Arp et al. [20], who recommend that the test data should chronologically follow the training data. Given that our E-PhishLLM was generated in 2025, and that all our considered datasets contained data collected much earlier (see Table 1), it follows that our assessment resembles a realistic evaluation. This difference leads to two deviations: for Experiment-1, we simply test all of our (already trained/fine-tuned) models on E-PhishLLM; for Experiment-2, we train/fine-tune each model on all our 8 considered datasets, and then test it on E-PhishLLM. Finally, for Experiment-3, we expand the list of considered LLMs to cover 14 (up from 2) models, each tested on E-PhishLLM.

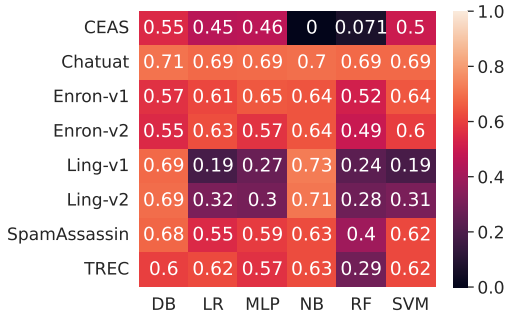
Results. We present the results of the experiments in Figure 4. We can highlight the following results:

- *Experiment-1*: the cross-evaluation settings are very challenging for all of our models, with F1-score ranging from 0 (for the NB trained on CEAS) to 0.73 (for the NB trained on Ling-v1). The DB model has the most consistently-high F1-score (ranging 0.55–0.71), and the Chataut dataset also leads to the highest overall F1-score (ranging 0.69–0.71).

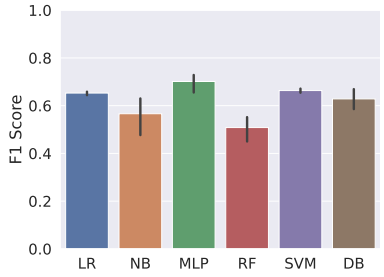
- *Experiment-2*: the all-vs-one setting shows a similar trend, i.e., the F1-score—while higher—does not go above 0.75. This indicates that our E-PhishLLM dataset is substantially different from any of our eight considered datasets.
- *Experiment-3*: intriguingly, LLMs show quite high and robust detection results, with 12 (out of 14) of the models exhibiting an F1-score ≥0.8. Some models, like `claude-3.5-haiku` shows brilliant performance, with an F1-score ≈0.95; the worst is gpt-3.5-turbo (F1-score ≈0.7).

We report the complete results (standard deviations, as well as accuracy/precision/recall) in our repository [11].

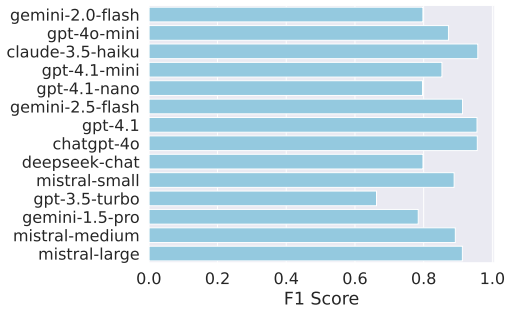
Answer to RQ4. Detectors trained on any combinations of our eight considered datasets struggle to detect the phishing emails in our E-PhishLLM dataset. LLMs, however, are much more effective. These results show that our proposed E-PhishLLM dataset represents a better "benchmark" than existing datasets to test previously proposed detectors.



(a) Experiment-1 (cross-Evaluation) English-only emails.



(b) Experiment-2 (all-vs-one). English-only.



(c) Experiment-3 (LLMs). English only emails.

Fig. 4: Benchmarking existing detectors on our proposed E-PhishLLM dataset.

5.2 User Study [RQ5]

To fairly compare the quality of the emails included in E-PhishLLM w.r.t. those included in existing datasets, we carried out a user study with 30 participants. To the best of our knowledge, we are the first to evaluate the quality of a phishing-email dataset in such a way.

Procedure. We used convenience sampling [37] to achieve a target number of 30 participants: each invitee was unaware of our research and had some background in cybersecurity. Each participant had to fill an online questionnaire, which began with some instructions and preliminary demographics questions. Then, each participant evaluated 20 phishing emails: 5 sampled from E-PhishLLM, and the remaining 15 sampled from SpamAssassin, Enron, Nazario[2] (5 each). We created 10 versions of the questionnaire, so each set of 20 emails was seen by 3 participants—this enabled us to cover 200 emails in total (50 per dataset). To evaluate an email, the participant had to rate the "overall phishing quality" of the email, i.e., how convincing, well-written, and realistic it appeared; the rating was expressed via a 1–5 likert scale (1=low, 5=high). The questionnaires are available in our repository [11] (snippets are in Figures 5, 6).

Sample Demographics and Results. Most participants were aged 18–25 (55%) or 26–40 (45%). Occupations included students (40%) and various roles in tech, consulting, and other sectors. Regarding IT expertise, 52% identified as intermediate, 22% as expert, and 16% as beginner. Cybersecurity experience was mostly intermediate (32%), with others reporting beginner (22%), no experience (23%), advanced (20%), or expert (3%). The average quality scores were: E-PhishLLM=3.41 (std=1.2); Nazario=2.65 (std=1.3); SpamAssassin=1.57 (std=1.0); Enron=1.45 (std=0.9). Even a t-test confirms that the perceived quality of the emails from E-PhishLLM is statistically superior to those of the other three datasets (p<.05).

Answer to RQ5. Yes, E-PhishLLM contains phishing emails of higher quality than those of SpamAssassin, Nazario, and Enron.

6 Discussion

Our open-problem paper should inspire a reflective exercise on research on phishing-email detection. Here, we state the limitations and ethics of our work, and then discuss the potential "dual use" of our proposed E-PhishGEN.

6.1 Limitations and Disclaimers

We transparently outline the limitations of our paper.

First, we acknowledge that our literature review (in §2.2) is not fully-systematic. Yet, this is not a problem because our conclusions (i.e., the fact that prior literature mostly focused on the same set of "overly-used" datasets) are shared also by prior work (see, e.g., [14, 29, 33, 47, 59, 61]). Carrying out a systematic literature review is outside the scope of this open-problem paper.

Second, a threat to validity of our reassessment (§3) is due to the potential labeling issues present in some of our considered datasets (in Table 1). We relied on the data provided by peer-reviewed prior work (including the Chataut dataset [30]). We carried out manual checks and we believe that the labeling in this dataset is not

[2]We considered SpamAssassin and Enron because we also used these in our assessment (§3); whereas we used Nazario as a representative of a popular dataset that has also more recent emails in it (see §2.2).

foolproof. However, we refrain from changing the ground truth of previously-proposed datasets: we release all of our code (including our random seeds) so future work can reproduce our experiments on hypothetical "fixed" versions of our considered datasets.

Third, for our user study (§5.2), we primed our users by mentioning that the study was about phishing: this may induce bias in the responses, but is a common practice in phishing studies [23] and we refrained from using deception. Moreover, also related to our user study, we only considered a small sample of each of our considered datasets, and we only solicited the opinion of 30 participants—so we acknowledge that our investigation cannot cover all cases.

Finally, **we do not assert** that not releasing source code or using private dataset diminishes the contributions of prior work. Our reassessment is a way to strengthen prior work's contributions, since our intra-dataset results confirm the findings of prior literature.

6.2 Ethical Considerations

Our institutions do not mandate that a formal IRB process is required to carry out the research discussed in this work. Still, we followed established ethical guidelines to carry out our research [22]. No human was harmed as a result of our user study, we did not use any deception, and we explicitly asked for each subject's consent to participate in our study. We also gave our contacts so that participants could ask us to delete their responses, if they so desire. The questionnaire was anonymous, and participation was voluntary and we offered no compensation. With regards to our proposed E-PhishGEN framework, we acknowledge that parts of it can be exploited by malicious entities (e.g., to generate phishing emails). Yet, we believe that the risk is minimal: real attackers *are well-aware* that LLMs can be used to craft phishing emails at scale [8], and we believe that the publication of this paper (and of its resources) would not aggravate this risk.

6.3 E-PhishGEN Dualism: Benign and Offensive AI

The E-PhishGEN exemplifies a dual-use technology with applications in both defense and offense.

Benign Usage: Enhancing Cyber Defense. E-PhishGEN can be used to improve phishing detection by generating synthetic data based on emerging phishing techniques. Cybersecurity experts can analyze novel attack patterns and then use the framework to create simulated phishing emails to fine-tune defense models. In addition, specialized data sets can be generated in various languages to improve the robustness of detection systems against diverse threats. The modularity of the framework allows companies to react to new threats quickly: if a novel phishing tactic emerges, companies can directly generate relevant company and employee profiles, define the characteristics of the new attack, and use the framework to simulate and incorporate these patterns into their internal detection systems, without the need to wait for real phishing attempts to be collected. This flexibility accelerates defense adaptation.

Malicious Usage: Empowering Offensive AI. Attackers can exploit OSINT (e.g., company websites, LinkedIn profiles) to gather detailed information about an organization. This data is fed into the framework's first module to create accurate employee profiles. Using the second module, attackers can generate customized spear phishing emails that mimic organizational communication, lowering the expertise barrier for sophisticated attacks. Thus, **E-PhishGEN** serves as a tool for *Offensive AI* [60], making phishing attacks more accessible and effective.

7 Recommendations and Future Work

We showed that research on phishing-email detection is relatively stagnant due to the overuse of (outdated) benchmark datasets.

Our contributions serve as a scaffold to revitalize the domain of phishing-email detection. Our open-source reassessment, reliant on cross-evaluation experiments, can be used to create new testbeds by future work. Our proposed E-PhishGEN framework can facilitate the generation of novel datasets—which can augment our proposed E-PhishLLM dataset. All such datasets, alongside being usable to test "generic" detectors of phishing emails, can also be used to devise "specific" detectors of LLM-generated emails—which is a subtle threat for which no solution exists.

We recommend future research to think deeply before using any given dataset for phishing-email detection to test their proposed methods. At the very least, the objective should be clearly stated: is it to "outperform previously proposed methods" or to "develop a method that can detect phishing emails in the real-world"?

Acknowledgments

The authors thank the AISec reviewers for the great feedback. Parts of this research has been funded by the Hilti Foundation.

References

[1] 2005. MillerSmiles. https://millersmiles.co.uk/archives/ Last update: 2025.
[2] 2005. Untroubled dataset. https://untroubled.org/spam/ Last Update: 2025.
[3] 2008. CEAS corpus. https://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/fooceas.
[4] 2018. IWSPA Dataset. https://dasavisha.github.io/IWSPA-sharedtask/.
[5] 2023. PhishBowl Dataset. https://it.cornell.edu/phish-bowl Last update: 2025.
[6] 2023. PhishingPot. https://github.com/rf-peixoto/phishing_pot Updated: 2025.
[7] 2024. *Interet Crime Report.* Technical Report. Federal Bur. of Investigation.
[8] 2024. *State of the Phish 2024.* Technical Report. ProofPoint. https://www.proofpoint.com/it/resources/threat-reports/state-of-phish.
[9] 2025. *IR Trends Q1 2025: Phishing soars as identity-based attacks persist.* Technical Report. Cisco Talos. https://blog.talosintelligence.com/ir-trends-q1-2025/.
[10] 2025. *Phishing Trends Report.* Technical Report. HoxHunt. https://hoxhunt.com/guide/phishing-trends-report.
[11] 2025. Repository of our paper. https://github.com/pajola/e-phishGen
[12] Sahar Abdelnabi, Katharina Krombholz, and Mario Fritz. 2020. Visualphishnet: Zero-day phishing website detection by visual similarity. In *ACM CCS*.
[13] Abdulla Al-Subaiey, Mohammed Al-Thani, Naser Abdullah Alam, Kaniz Fatema Antora, Amith Khandakar, and SM Ashfaq Uz Zaman. 2024. Novel interpretable and robust web-based AI platform for phishing email detection. *Computers and Electrical Engineering* (2024).
[14] Abeer Alhuzali, Ahad Alloqmani, Manar Aljabri, and Fatemah Alharbi. 2025. In-Depth Analysis of Phishing Email Detection: Evaluating the Performance of Machine Learning and Deep Learning Models Across Multiple Datasets. *Applied Sciences* (2025).
[15] Giovanni Apruzzese, Hyrum S Anderson, Savino Dambra, David Freeman, Fabio Pierazzi, and Kevin Roundy. 2023. "Real attackers don't compute gradients": bridging the gap between adversarial ML research and practice. In *IEEE SaTML*.
[16] Giovanni Apruzzese, Pavel Laskov, and Johannes Schneider. 2023. Sok: Pragmatic assessment of machine learning for network intrusion detection. In *IEEE EuroS&P*.
[17] Giovanni Apruzzese, Pavel Laskov, and Aliya Tastemirova. 2022. SoK: The impact of unlabelled data in cyberthreat detection. In *IEEE EuroS&P*.
[18] Giovanni Apruzzese, Luca Pajola, and Mauro Conti. 2022. The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE TNSM* (2022).
[19] APWG. 2024. *Phishing Activity Trends Report, Q3.* Technical Report. APWG.
[20] Daniel Arp, Erwin Quiring, Feargus Pendlebury, Alexander Warnecke, Fabio Pierazzi, Christian Wressnegger, Lorenzo Cavallaro, and Konrad Rieck. 2022. Dos and don'ts of machine learning in computer security. In *USENIX SEC*.
[21] Samer Atawneh and Hamzah Aljehani. 2023. Phishing email detection model using deep learning. *Electronics* (2023).

[22] Michael Bailey, David Dittrich, Erin Kenneally, and Doug Maughan. 2012. The menlo report. *IEEE Security & Privacy* (2012).
[23] Shahryar Baki and Rakesh M Verma. 2023. Sixteen years of phishing user studies: What have we learned? *IEEE TDSC* (2023).
[24] Lucas Betts, Robert Biddle, Danielle Lottridge, and Giovanni Russello. 2024. Exploring Content Concealment in Email. In *APWG eCrime*.
[25] Alp Barış Beydemir, Ulaş Sezgin, Umutcan Doğan, Burak Engin Aşıklar, Fahri Anıl Yerlikaya, and Şerif Bahtiyar. 2024. A Dynamically Selected GPT Model for Phishing Detection. In *IEEE ACIT*.
[26] Battista Biggio and Fabio Roli. 2018. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition* (2018).
[27] Panagiotis Bountakas, Konstantinos Koutroumpouchos, and Christos Xenakis. 2021. A comparison of natural language processing and machine learning methods for phishing email detection. In *ARES*.
[28] Panagiotis Bountakas and Christos Xenakis. 2023. Helphed: Hybrid ensemble learning phishing email detection. *JISA* (2023).
[29] Arifa I Champa, Md Fazle Rabbi, and Minhaz F Zibran. 2024. Curated datasets and feature analysis for phishing email detection with machine learning. In *IEEE International Conference on Computing and Machine Intelligence*.
[30] Robin Chataut, Prashnna Kumar Gyawali, and Yusuf Usman. 2024. Can ai keep you safe? a study of large language models for phishing detection. In *IEEE CCWC*.
[31] Robin Chataut, Yusuf Usman, Chowdhury Mohammad Abid Rahman, Sohan Gyawali, and Prashnna K Gyawali. 2024. Enhancing Phishing Detection with AI: A Novel Dataset and Comprehensive Analysis Using Machine Learning and Large Language Models. In *IEEE UEMCON*.
[32] Gordon V Cormack and Thomas R Lynam. 2005. TREC corpus. https://plg.uwaterloo.ca/cgi-bin/cgiwrap/gvcormac/foo07.
[33] Avisha Das, Shahryar Baki, Ayman El Aassal, Rakesh Verma, and Arthur Dunbar. 2019. SoK: a comprehensive reexamination of phishing research from the security perspective. *IEEE Communications Surveys & Tutorials* 22, 1 (2019), 671–708.
[34] Sara de Rosa, Francesco Gringoli, and Gabriele Bellicini. 2024. Hey ChatGPT, Is This Message Phishing?. In *IEEE MedComNet*.
[35] Rachna Dhamija, J Doug Tygar, and Marti Hearst. 2006. Why phishing works. In *ACM CHI*.
[36] Jay Doshi, Kunal Parmar, Raj Sanghavi, and Narendra Shekokar. 2023. A comprehensive dual-layer architecture for phishing and spam email detection. *Computers & Security* (2023).
[37] Ilker Etikan, Sulaiman Abubakar Musa, Rukayya Sunusi Alkassim, et al. 2016. Comparison of convenience sampling and purposive sampling. *American journal of theoretical and applied statistics* (2016).
[38] Yong Fang, Cheng Zhang, Cheng Huang, Liang Liu, and Yue Yang. 2019. Phishing email detection using improved RCNN model with multilevel vectors and attention mechanism. *IEEE Access* (2019).
[39] Apache Foundation. 2005. SpamAssassin Corpus. https://spamassassin.apache.org/old/publiccorpus/.
[40] Yanick Fratantonio, Luca Invernizzi, Loua Farah, Kurt Thomas, Marina Zhang, Ange Albertini, Francois Galilee, Giancarlo Metitieri, Julien Cretin, Alex Petit-Bianco, David Tao, and Elie Bursztein. 2025. MAGIKA: AI-Powered Content-Type Detection. In *ACM ICSE*.
[41] Surajit Giri, Siddhartha Banerjee, Kunal Bag, and Dipanjan Maiti. 2022. Comparative study of content-based phishing email detection using global vector (GloVe) and bidirectional encoder representation from transformer (BERT) word embedding models. In *IEEE ICEEICT*.
[42] Payas Gupta, Bharat Srinivasan, Vijay Balasubramaniyan, and Mustaque Ahamad. 2015. Phoneypot: Data-driven understanding of telephony threats.. In *NDSS*.
[43] Reeber Erik Forman George Hopkins, Mark and Jaap Suermondt. 1999. Spambase. https://doi.org/10.24432/C53G6X.
[44] Suhaima Jamal, Hayden Wimmer, and Iqbal H Sarker. 2024. An improved transformer-based model for detecting phishing, spam and ham emails: A large language model approach. *Security and Privacy* (2024).
[45] Bryan Klimt and Yiming Yang. 2004. The enron corpus: A new dataset for email classification research. In *ECML*.
[46] Deeksha Kulal, Leul Shiferaw, and Quamar Niyaz. 2025. Phishing Email Detection Through Machine Learning and Word Error Correction. In *IEEE COMSNETS*.
[47] Phyo Htet Kyaw, Jairo Gutierrez, and Akbar Ghobakhlou. 2024. A Systematic Review of Deep Learning Techniques for Phishing Email Detection. *Electronics* (2024).
[48] Daniele Lain, Tarek Jost, Sinisa Matetic, Kari Kostiainen, and Srdjan Capkun. 2024. Content, Nudges and Incentives: A Study on the Effectiveness and Perception of Embedded Phishing Training. In *ACM CCS*.
[49] Daniele Lain, Kari Kostiainen, and Srdjan Čapkun. 2022. Phishing in organizations: Findings from a large-scale and long-term study. In *S&P*.
[50] Jehyun Lee, Farren Tang, Pingxiao Ye, Fahim Abbasi, Phil Hay, and Dinil Mon Divakaran. 2021. D-fence: A flexible, efficient, and comprehensive phishing email detection system. In *IEEE EuroS&P*.
[51] Sakshi Mahendru and Tejul Pandit. 2024. Securenet: A comparative study of deberta and large language models for phishing detection. In *IEEE BDAI*.
[52] Parisa Mehdi Gholampour and Rakesh M Verma. 2023. Adversarial robustness of phishing email detection models. In *ACM IWSPA*.
[53] Ayat Najjar, Huthaifa I Ashqar, Omar Darwish, and Eman Hammad. 2025. Leveraging Explainable AI for LLM Text Attribution: Differentiating Human-Written and Multiple LLMs-Generated Text. *arXiv preprint arXiv:2501.03212* (2025).
[54] Jose Nazario. 2015. Nazario Email Corpus. https://monkey.org/~jose/phishing/.
[55] Quan Hong Nguyen, Tingmin Wu, Van Nguyen, Xingliang Yuan, Jason Xue, and Carsten Rudolph. 2024. Utilizing large language models with human feedback integration for generating dedicated warning for phishing emails. In *ACM Workshop on Secure and Trustworthy Deep Learning Systems*.
[56] Dragomir Radev. 2008. CLAIR collection of fraud email. https://aclweb.org/aclwiki/CLAIR_collection_of_fraud_email_(Repository).
[57] Sergio Rojas-Galeano. 2024. Zero-Shot Spam Email Classification Using Pre-trained Large Language Models. In *Workshop on Engineering Applications*.
[58] Georgios Sakkis, Ion Androutsopoulos, Georgios Paliouras, Vangelis Karkaletsis, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. 2003. A memory-based approach to anti-spam filtering for mailing lists. *Inf. retrieval* (2003).
[59] Said Salloum, Tarek Gaber, Sunil Vadera, and Khaled Shaalan. 2022. A systematic literature review on phishing email detection using natural language processing techniques. *IEEE Access* (2022).
[60] Saskia Laura Schröer, Giovanni Apruzzese, Soheil Human, Pavel Laskov, Hyrum S Anderson, Edward WN Bernroider, Aurore Fass, Ben Nassi, Vera Rimmer, Fabio Roli, et al. 2025. SoK: On the offensive potential of AI. In *IEEE SaTML*.
[61] Kutub Thakur, Md Liakat Ali, Muath A Obaidat, and Abu Kamruzzaman. 2023. A systematic review on deep-learning-based phishing email detection. *Electronics* (2023).
[62] Daniel Timko and Muhammad Lutfor Rahman. 2024. Smishing dataset i: Phishing sms dataset from smishtank.com. In *ACM CODASPY*.
[63] Marie Weinz, Nicola Zannone, Luca Allodi, and Giovanni Apruzzese. 2025. The Impact of Emerging Phishing Threats: Assessing Quishing and LLM-generated Phishing Emails against Organizations. In *ACM AsiaCCS*.
[64] Yinuo Xue, Eric Spero, Yun Sing Koh, and Giovanni Russello. 2025. MultiPhishGuard: An LLM-based Multi-Agent System for Phishing Email Detection. *arXiv:2505.23803* (2025).
[65] Peng Zhao and Shuyuan Jin. 2024. Fewshing: A few-shot learning approach to phishing email detection. In *IEEE SEAI*.
[66] Thomas H Zunder. 2021. A semi-systematic literature review, identifying research opportunities for more sustainable, receiver-led inbound urban logistics flows to large higher education institutions. *European Transport Research Review* (2021).

A User Study Snippets

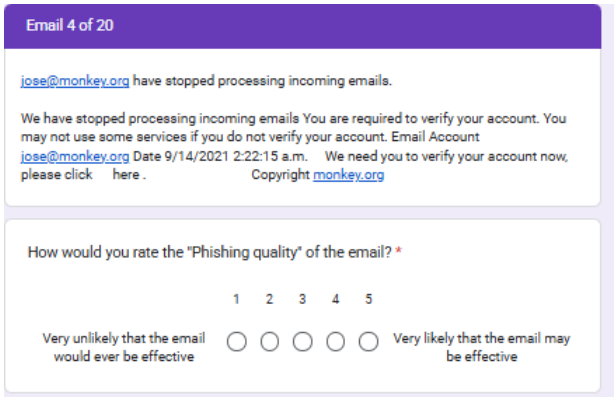


Fig. 5: Illustrative example of an email evaluated in our user study. The email is taken from the Nazario dataset.

B Post-acceptance Responses (Q&A)

The AISec reviewers provided great feedback. We would like to respond directly here to their suggestions/remarks, in a Q&A format.

Would using different LLMs to generate dataset influence detection performance? It depends. There is evidence that LLMs have different writing styles (e.g., [53]). Therefore, using a single LLM to generate a dataset would mean that any ML model trained on such a dataset would likely be effective at detecting emails generated by the same

LLM that generated the dataset. Given that GPT-based models are widespread, we used these models to create E-PhishLLM. However, we acknowledge that, for a truly comprehensive dataset that can cover the writing style of multiple LLMs, it would be desirable to expand our proposed E-PhishLLM by generating additional samples using different LLMs (which we leave to future work).

Could there be bias in the user study? Older datasets might reflect outdated phishing styles, while newer samples, shaped by current trends, could appear unfamiliar or more convincing simply because they differ from the "already known" types of phishing emails. In our user study, we asked a specific question: "How would you rate the 'phishing quality' of the email?". Our rationale is that, since our study is done in 2025, the rating provided by our participants would reflect the "quality" according to current trends. Note: it is implicit that, in claiming that E-PhishLLM has emails of "better quality", *we are specifically referring to the current state of phishing*. It would be false to state that the emails contained in previously proposed datasets to be of poor quality in the general sense (after all, those emails were *real* phishing emails—but most such emails pertain to outdated phishing tactics, since they were exchanged, e.g., years before the advent of LLMs).

You discussed the F1-score in the paper, can you provide more insights on the false positives? That's true. We focused on the F1-score because it is the only metric which accounts for both detection rate and false-positive rate. However, we agree that measuring false positives is a crucial metric for cyber-threat detection purposes. We provide additional metrics (recall, precision, accuracy) in our repository [11]. Moreover, and to get a broad overview of the effects of false positives, we report in Figure 7 the distribution of the precisions (i.e., $\frac{TP}{TP+FP}$) obtained by aggregating the results of *all* our models in the re-assessment experiments (in §3); intuitively, a precision close to 1 indicates a low number of false positives. We see that the precision is very high in the baseline case (i.e., when a model is trained/fine-tuned and tested on data from the same "source" dataset), but it drops substantially when the test is done on a different dataset. Note that we do not fine-tune the LLMs, which is why they are placed in a different category.

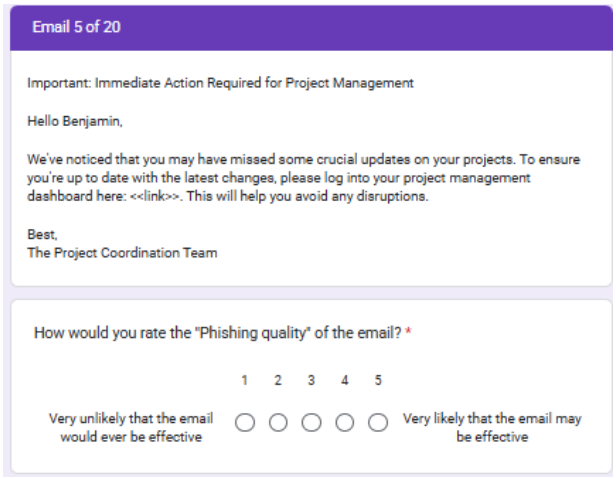


Fig. 6: Illustrative example of an email evaluated in our user study. The email is taken from the E-PhishLLM dataset.

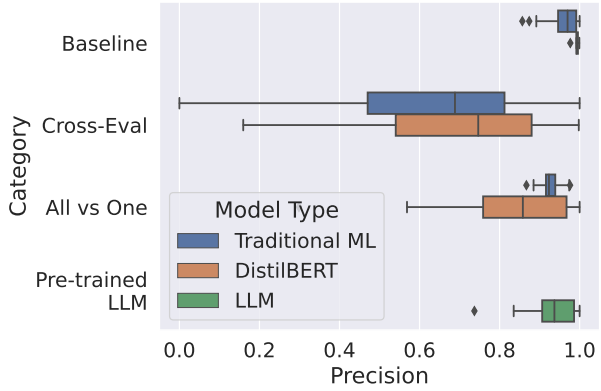


Fig. 7: Precision results of our reassessment experiments. Each boxplot represents the distribution of the precision ($\frac{TP}{TP+FP}$) across all of our experiments on existing datasets. For instance, the blue bins show the aggregated results of all the models using feature-based techniques.

Does dataset balancing matter? The dataset we used in our re-assessment, according to Table 1, have different compositions: some are balanced (e.g., Enron-v1) others are very unbalanced (e.g., Ling-v1). Yet, the same-dataset performance is always high even in highly-unbalanced cases. It is difficult to draw absolute conclusions in the cross-evaluation case, as performance varies even when the models are trained on imbalanced datasets. For instance, the MLP trained on Ling-v2 (highly unbalanced) has 0.53 F1-score on CEAS, but 0.79 on SpamAssassin; whereas the MLP trained on Enron-v2 (balanced) has 0.73 F1-score on CEAS, but 0.42 on SpamAssassin. We believe that, rather than "balancing", the major difference is the type of phishing email contained in each of these datasets.

The literature review does not include approaches from the Industry. That is true. We did not cover these, because our literature review was rooted on academic work. (Recall the point of departure of our research was: "what are the datasets commonly used in phishing-email-detection literature?" Informally, we hope that, if ML is used in industry, the corresponding methods are trained on "better" datasets than those used in academic literature!) We therefore acknowledge that our analysis does not include approaches for industry. This could be a room for future work.

What about hyperparameter configurations? We tried to align our reassessment to prior work. This is why we configured our models by using the parameters reported by prior work which have been found to work better in the specific context of phishing email detection. Given that our results align to those claimed by prior work, we believe our choice to be valid and functional to our purpose.

A valuable future direction would be to integrate E-PhishLLM-generated emails into a complete phishing campaign tool for controlled testing with real users. This would offer a realistic validation of the dataset's effectiveness. This is a wonderful suggestion that we will certainly consider pursuing.