Curated Datasets and Feature Analysis for Phishing Email Detection with Machine Learning

Arifa I. Champa Md Fazle Rabbi Minhaz F. Zibran
Department of Computer Science, Idaho State University
Pocatello, ID, USA
{arifaislamchampa, mdfazlerabbi, minhazzibran}@isu.edu

2024 IEEE 3rd International Conference on Computing and Machine Intelligence (ICMI) | 979-8-3503-7297-7/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICMI60790.2024.10585821

***Abstract*—Despite continued research, phishing email attacks are on the rise and there is a lack of rich curated datasets for training and testing email filtering techniques. To address this, we produce and release seven curated datasets with 203,176 email instances for use with machine learning (ML) to distinguish phishing emails from legitimate ones. We create these datasets by meticulously curating phishing and legitimate emails from different repositories. Then to demonstrate that our curated datasets are suitable for the purpose, we conduct a quantitative analysis for evaluating the performance of five ML algorithms. We also analyze the significance and impact of different features within these curated datasets on those ML algorithms. These curated datasets along with the findings from the quantitative analysis will advance the development of a robust defense against phishing attacks.**

***Index Terms*—Phishing email, data curation, natural language processing, machine learning, quantitative analysis, detection**

I. Introduction

For several decades email communication has been an integral part of both professional and personal life. However, this has also made it a prime target for phishing attacks, which pose significant security risks. Phishing email attacks involve cybercriminals creating deceptive emails that mimic legitimate correspondence to trick unsuspecting users into revealing sensitive information. These tactics can lead to data breaches and various malicious activities. Surprisingly, it is not just common individuals who are susceptible; even individuals with advanced education can fall victim to these schemes [1]. Shockingly, it is estimated that around 91% of hacking attempts start with a phishing email [2], and an astonishing 3.4 billion phishing emails are sent daily [3].

This paper addresses the growing concern of phishing attacks, which are becoming more frequent and sophisticated. In 2023 alone, nearly 6 billion security breaches occurred, with phishing attacks occurring at a rate of approximately one every 11 seconds [3]. Despite technological advancements, phishing remains a significant threat, with cybercriminals continuously devising new scams. For instance, Gmail's filters intercept millions of phishing emails, with a significant portion of them being new and previously unseen scams [4]. These alarming statistics underscore the pressing need for more effective and robust methods of detecting phishing attempts.

In light of this critical necessity, phishing attacks have been a subject of extensive research [5]–[8]. A significant challenge in this area of study is the scarcity of real-life, diverse, well-curated, rich datasets [9]. This piqued our interest to take a thorough look into the suitability of the potential email repositories/datasets for phishing email detection. A key realization from this investigation is that existing email repositories/datasets, while potentially valuable, are not immediately usable due to several reasons. Some repositories include raw emails which must be first curated before those can be fed to machine learning (ML) algorithms. Some emails are encoded/encrypted, which need to be decoded/decrypted first; some are HTML-formatted, from which plain-text content must be extracted first, some emails are written in languages other than emails while some come with empty bodies, which must be filtered before ML algorithms can be applied. Above all, curation with extraction of features such as URLs is essential before they become ready for ML algorithms.

We, therefore, create and publicly release seven phishing email datasets meticulously curated to make them ready for the application of ML algorithms. These datasets include collections of phishing and legitimate samples of curated emails representing real-world emails ranging from personal to corporate communications across different times in history. We also recognize that for the pragmatic implementation and adoption of ML-based solutions for phishing email detection, it is of the utmost importance to identify the characteristics and attributes of email communications that contribute most to the distinction of phishing emails from legitimate ones. Thus, we also apply five prominent ML algorithms on the datasets, first, to verify and demonstrate the suitability of our curated datasets in ML applications, and second, to identify the most influential email features that impact most on the ML algorithms' classification of phishing and non-phishing emails.

Major Contributions: To be clear and specific, we reiterate the two major contributions of this work as enumerated below.

- We release a collection of seven datasets [10], [11], each with a balanced collection of phishing and legitimate emails, meticulously curated and ready for immediate application of ML algorithms or similar analyses.
- Applying five prominent ML algorithms on the datasets, we demonstrate the datasets' suitability and readiness for ML applications, while we also derive insights into the impacts of different email features on the ML algorithms' decision-making. These insights are essential for the development of sophisticated and targeted approaches for the purpose.

The rest of the paper is organized as follows. In Section II, we describe our methodology for the curation and creation of seven datasets. Section III describes the resultant curated datasets. In Section IV, we present a quantitative analysis, where we apply five ML algorithms to demonstrate the suitability of our datasets and to examine feature importance. Section V addresses the limitations of this study and our efforts to minimize them. In Section VI, we discuss the work in the literature related to ours. Finally, Section VII concludes the paper with some future research directions.

II. Curation and Creation of Datasets

As portrayed in Figure 1, this work is carried out in two phases. In phase-1, we process and curate the collections of emails to produce seven datasets. In phase-2, we apply five ML algorithms to the datasets to demonstrate their suitability and to derive insights into feature importance. The procedural steps involved in each of the phases are summarized in Figure 1, and described in Section II-B and Section IV.

A. Sources of Email Collections

As mentioned before, we collect emails from several sources. The Ling-Spam (Ling) [12] repository includes emails from the year 2000, representing early email communication. These emails specifically focused on topics of interest to linguists. The Enron [13] corpus includes emails dated back to the year 2006. The Apache SpamAssassin repository (Assassin) [14] includes emails spanning over multiple years (2002 - 2006). The TREC public corpus periodically released in 2005 (TREC-05) [15], 2006 (TREC-06) [16], and 2007 (TREC-07) [17] include various email communications in consecutive three years. The CEAS 2008 Challenge Lab Evaluation Corpus (CEAS-08) [18] includes emails dating back to the year 2008.

B. Processing for Curation

Although the emails from all the aforementioned sources are distinguished into phishing or legitimate, they are mostly raw and need varied levels of processing to be prepared for ML applications. Hence we further process and curate them to create our datasets. The activities involved in data curation are summarized as phase-1 in Figure 1 and elaborated below.

1) Decoding: The emails in all the repositories, except for Ling, contain encoded/encrypted emails that need to be decoded/decrypted first. To decode these, we first parse the raw content to identify the type of encoding, as specified in the email headers. We then apply different decoding techniques for different types of encoded data. For example, the process of decoding Base64 encoded data involves reversing the Base64 encoding process. Then, we address different character sets, as guided by the Content-Type header. This ensures that the decoded byte sequence is accurately translated into a readable string. This overall decoding method transforms the encoded segments of an email back into their original, comprehensible format, which is crucial for their continuous integration and analysis in our research. Here, we utilize the Python `email` library [19] to assist in the decoding of encoded emails.

2) Extraction of Plain Text: We observe that within all repositories, with the exception of Ling, there is a mix of HTML-formatted emails and plain text emails. In order to maintain consistency and ensure a dataset feedable to ML algorithms, we extract plain text from the HTML-formatted emails. At first, we retain the content within certain formatting tags, such as and <p>, removing only the tags themselves to preserve essential data without the visual formatting. Then, consecutive newlines are replaced with single spaces to enhance machine readability by eliminating unnecessary breaks. The extracted plain text allows more straightforward and more consistent data processing, building a robust foundation for subsequent phases of the work.

3) Duplicate Removal: To identify duplicate emails, we analyze the 'Body' content of each email, which is the crucial part of an email. If two emails share identical 'Body' content, we classify them as duplicates. Then, we remove the redundant emails by keeping single instance from each duplicate group.

4) Discrepancy Handling: Within the selected repositories, we observe that certain emails have empty email bodies. We remove such emails from our study. Additionally, we keep only emails written in English to maintain consistency.

5) Data Cleansing: To improve the integrity and quality of the dataset, we apply data cleansing. This involves removing stop words, such as 'and', 'the', and 'is', from each email. This step is crucial as it minimizes noise in the dataset.

6) Vectorization: The email data must be transformed into numerical vector representations for feeding to ML algorithms. To accomplish this transformation, we utilize term frequency-inverse document frequency (TF-IDF) technique from natural language processing (NLP) which is the most widely used statistical method [20]–[22]. This method involves multiplying term frequency (TF) of a word by its inverse document frequency (IDF). Term frequency signifies how frequently a word appears in a particular document, while inverse document frequency denotes the prevalence of the word across the entire document collection. The 'Urls' attribute is transformed into a binary feature: a value of 1 indicates the presence of URL(s) in the email body, whereas 0 indicates their absence.

III. Resultant Curated Datasets

Upon completion of data curation as described above, we obtain the curated datasets corresponding to the seven original repositories and we name each dataset after the name of the original repository it is derived from. Table I provides a summary of these seven curated datasets we produce. The topmost row in this table identifies each of the datasets.

For each dataset, the second, third, and fourth row from the top respectively include the number of email instances processed at decoding, duplicate removal, and discrepancy handling. The remaining rows in Table I present an overview of the resultant datasets after the completion of data curation. Thus the subsequent row include the total number of curated emails, the number of legitimate emails, the number of scam/phishing emails, the ratio of legitimate to scam emails (Legit:Scam), and the features available in each dataset.

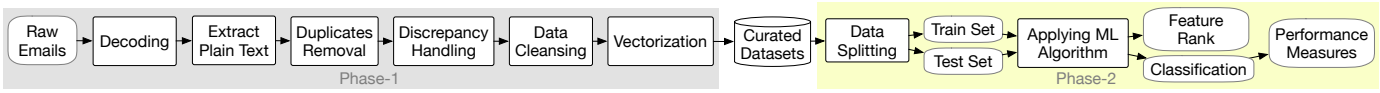

Fig. 1. Procedural steps at different phases in our work

For example, we obtain 30,494 emails upon completion of decoding Enron dataset, from which 724 duplicates are removed, and upon handling discrepancies in three emails in the remaining, we obtain a total of 29,767 curated emails. 15,791 of these curated emails are legitimate, while the rest 13,976 are phishing emails resulting in a legit:scam ratio of 53:47. As shown in the bottom row, the curated Enron dataset includes only two email features: 'Subject' and 'Body'. Only Ling and Enron datasets have only two features, i.e., 'Subject' and 'Body', while other datasets include features, which are 'Sender', 'Receiver', 'Date', 'Subject', 'Body', and 'Urls'.

TABLE I
SUMMARY OF CURATED DATASETS WE HAVE CREATED AND RELEASED

Dataset	Ling	Enron	Assassin	TREC-05	TREC-06	TREC-07	CEAS-08
Decoded	0	30,494	6,047	92,188	37,786	75,417	1,37,701
Duplicates	34	724	220	29,500	20,079	19,026	70,100
Discrepancy	0	3	18	3,413	254	3	1,217
Total	2,859	29,767	5,809	55,414	16,416	53,757	39,154
Legitimate	2,401	15,791	4,091	32,329	12,411	24,358	17,312
Scam	458	13,976	1,718	23,085	4,005	29,399	21,842
Legit:Scam	84:16	53:47	70:30	58:42	76:24	45:55	44:56
Features	Subject, Body	Sender, Receiver, Date, Subject, Body, Urls					

As observed in Table I, a large number of duplicate emails are removed in the curation of TREC-05, TREC-06, TREC-07, and CEAS-08 datasets. Additionally, the discrepancy handling step impacts the TREC-05 and then CEAS-08 contributing to a significant reduction in the number of email instances. However, this removal of inconsistencies is necessary for the robustness of ML algorithms. Finally, after the curation is completed, the TREC-05 and TREC-07 datasets contain more than 50 thousand email instances, while CEAS-08 and Enron have nearly 40 thousand and 30 thousand emails respectively.

As seen in Table I, the curated datasets Enron, TREC-07, and CEAS-08 are the fairly balanced in terms of phishing and legitimate email ratios. On the contrary, the curated Ling dataset is the most imbalanced with a Legit:Scam ratio of 84:16. The TREC-06 and Assassin datasets also appear to be imbalanced. Although ML algorithms generally perform better with balanced datasets, in real-world situations, particularly in phishing email detection, ML algorithms must deal with adverse circumstances and imbalanced datasets. Thus, our release [10] of the curated datasets includes both balanced and imbalanced datasets of diverse sizes to pose pragmatic challenges to ML algorithms or similar analyses for making the target solutions robust and reliable.

IV. APPLYING ML TO OUR CURATED DATASET

Having created the seven curated datasets, we want to determine whether or not they are suitable for the application of ML algorithms and how well the ML algorithms in association with NLP techniques perform in the detection of phishing emails when operated on our curated datasets.

A. Procedure

1) Choice of ML Algorithms: To identify phishing emails, we apply five distinct ML algorithms: Support Vector Machine (SVM), Random Forest (RF), Extra Tree (ET), XGBoost (XGB), and AdaBoost (ADB). These algorithms are chosen for their wide recognition and effectiveness in phishing email detection [20]–[23]. Additional information about these chosen ML algorithms can be found elsewhere [24].

2) Splitting into Training and Test Samples: Instead of opting for a straightforward train-test split, we adopt "stratified k-fold", which is a more robust approach for implementing 10-fold cross-validation while preserving the percentage of samples for each class. This approach ensures a balanced representation of different classes in both our training and testing datasets, closely mirroring the distribution found in the complete dataset. Such a method is particularly suitable for handling datasets with imbalanced class distributions, such as Ling, Assassin, and TREC-06 in our curated datasets.

3) Metrics for Evaluation: For every ML algorithm applied to each dataset, we document the number of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). In our study, FN indicates the number of phishing email incorrectly identified as legitimate, while FP represents the number of legitimate email incorrectly identified as phishing. In contrast, TP occurs when the algorithm correctly labels a phishing email as phishing. Similarly, TN is the case where the algorithm correctly identifies legitimate emails. We then evaluate the performance of each algorithm using metrics like accuracy, recall, precision, F-score, and ROC [20].

4) Computation for Feature Importance: Feature importance calculation in ML varies depending on the algorithm used. In SVM, each feature has a coefficient which represents its weight in the decision function. Feature importance is calculated by taking the absolute value of these coefficients. A larger absolute value indicates that the feature has a greater impact on the decision boundary. In ensemble models such as RF and ET, feature importance is calculated by measuring how each feature contributes to reducing Gini impurity [25] across all the trees in the forest. The values from each tree are then averaged to determine overall feature importance. In ADB, feature importance is determined by evaluating how well each feature helps decrease prediction errors as weak learners are combined in a sequence. In XGB, feature importance is determined by summing the gains, which show how effectively each feature boosts performance by reducing errors. This calculation considers all instances where the feature is used across all trees in the ensemble.

After finding the feature importance of each feature across all the curated datasets, we normalize them to a range between 0.0 to 1.0. This enables a fair comparison among the feature importances in the selected five ML algorithms. For each

algorithm operated on a certain dataset, we assign ranks to the features in the dataset according to a descending order of normalized feature importance scores achieved by the ML algorithm operated on the dataset.

5) Invoking ML Algorithms: For applying ML algorithms to detect phishing emails in each of the curated datasets, we use the 'Subject' and 'Body' features for Ling and Enron datasets, as for the remaining five datasets, we utilize the 'Sender', 'Receiver', 'Date', 'Subject', 'Body', and 'Urls' features. For each curated dataset, the training subset is used to train each of the chosen the five ML algorithms utilizing their default hyper-parameters. Finally, the testing subsets are used to evaluate the performance of the algorithms. For each ML algorithm operated on each dataset, we compute the evaluation metrics, normalized feature importance scores, and feature ranks described above.

B. Analysis and Findings

For each of the curated datasets, the performances of the five ML algorithms, as measured in accuracy, precision, recall, and F-score (averaged over 10 runs for the 10-fold cross validation) are presented in Table II. For each curated dataset, the cells shaded in green indicate the best values achieved for the metrics, while the cells shaded in red indicate the worst metric values achieved by the five ML algorithms.

TABLE II
PERFORMANCE OF ML ALGORITHMS ON OUR CURATED DATASETS

Dataset	Metrics	SVM	RF	ET	XGB	ADB
Ling	Accuracy (%)	84.21	97.19	97.54	98.95	98.25
	Precision (%)	70.91	97.28	97.61	98.96	98.23
	Recall (%)	84.21	97.19	97.54	98.95	98.25
	F-score (%)	76.99	97.08	97.46	98.93	98.24
Enron	Accuracy (%)	95.87	98.69	98.69	98.45	95.70
	Precision (%)	96.09	98.69	98.69	98.46	95.72
	Recall (%)	95.87	98.69	98.69	98.45	95.70
	F-score (%)	95.87	98.69	98.69	98.45	95.70
Assassin	Accuracy (%)	95.34	98.28	98.45	98.45	98.79
	Precision (%)	95.52	98.28	98.45	98.45	98.79
	Recall (%)	95.34	98.28	98.45	98.45	98.79
	F-score (%)	95.24	98.27	98.44	98.45	98.79
TREC-05	Accuracy (%)	97.47	98.86	99.12	98.57	95.51
	Precision (%)	97.47	98.86	99.12	98.57	95.51
	Recall (%)	97.47	98.86	99.12	98.57	95.51
	F-score (%)	97.47	98.86	99.12	98.57	95.51
TREC-06	Accuracy (%)	93.48	96.34	97.01	97.99	95.43
	Precision (%)	93.78	96.44	97.10	97.99	95.39
	Recall (%)	93.48	96.34	97.01	97.99	95.43
	F-score (%)	93.17	96.25	96.95	97.97	95.40
TREC-07	Accuracy (%)	99.33	99.78	99.85	99.80	98.33
	Precision (%)	99.33	99.78	99.85	99.80	98.36
	Recall (%)	99.33	99.78	99.85	99.80	98.33
	F-score (%)	99.33	99.78	99.85	99.80	98.33
CEAS-08	Accuracy (%)	97.57	99.62	99.69	99.64	97.55
	Precision (%)	97.58	99.62	99.69	99.64	97.61
	Recall (%)	97.57	99.62	99.69	99.64	97.55
	F-score (%)	97.57	99.62	99.69	99.64	97.55

1) Performance of ML Algorithms: As observed in Table II, for the comparatively balanced datasets Enron, TREC-07, CEAS-08, and TREC-05, ET performs the best while ADB performs the worst.

This scenario changes for the highly imbalanced datasets, namely Assassin, Ling, and TREC-06. For these datasets, the boosting algorithms ADB and XGB appear superior. Their strategy of progressively focusing on misclassified data points allows them to enhance the accuracy for the lesser-represented class. It is interesting to see that, for the highly imbalanced datasets, SVM clearly falls behind the four ensemble models RF, ET, ADB, and XGB. To gain a deeper understanding of the specific reasons behind this underperformance, conducting a feature importance analysis is essential. The insights obtained from this feature analysis, as discussed in Section IV-B2, help explain the underperformance of SVM.

For the large datasets, particularly those with nearly 40K instances or more such as TREC-05, TREC-07, and CEAS-08, ET consistently outperforms all other ML algorithms. This superior performance can be attributed to the unique randomization features and the capability to adeptly navigate and manage the complexities of feature interactions within the data of ET. In contrast, for these large datasets, ADB performs poorer than all other ML algorithms. However, ADB performs better when applied on smaller datasets Ling and Assassin.

TABLE III
CONFUSION MATRIX VALUES FOR OUR CURATED DATASETS

Dataset	Best ML Algorithm	TP	TN	FP	FN	F (%)
TREC-07	ET	2,938	2,429	7	1	0.15
CEAS-08	ET	2,176	1,727	4	8	0.31
Assassin	ADB	167	809	3	4	0.71
TREC-05	ET	2,276	3,214	18	33	0.92
Ling	XGB	42	240	0	3	1.05
Enron	ET	1,377	1,560	19	20	1.31
TREC-06	XGB	374	1,234	7	26	2.01

For each of our seven curated datasets, the best-performing ML algorithm, along with the corresponding number of TP, TN, FP, and FN are presented in Table III. The false prediction (F) percentages of the best-performing ML algorithms are also included in the rightmost column of Table III. As evident from Table III, the lowest percentage of false predictions are found for TREC-07 (0.15%) and CEAS-08 (0.31%).

Across all curated datasets, each of the five ML algorithms achieves high ROC scores (98.55% or above). For example, in Figure 2, we present the ROC curves for these algorithms applied to the Enron dataset where ET performs the best with an ROC score of 99.88%. Similar ROC curves with high scores are observed for all the curated datasets. These high ROC scores as well as the high accuracy, precision, recall, and F-scores (Table II) achieved by the ML algorithms on all the curated datasets imply that our datasets are well curated allowing the ML algorithms to perform very well in email classification.

2) Feature Importance Analysis: As discussed in Section IV-B1, some ML algorithms perform better in relatively large, balanced datasets while other algorithms are found to have performed better in relatively smaller imbalanced datasets. To understand the reasons and to derive insights into the relative importance of different email features, we carry out a feature analysis.

For each of the datasets, the normalized feature importance scores captured for each of the five ML algorithms are

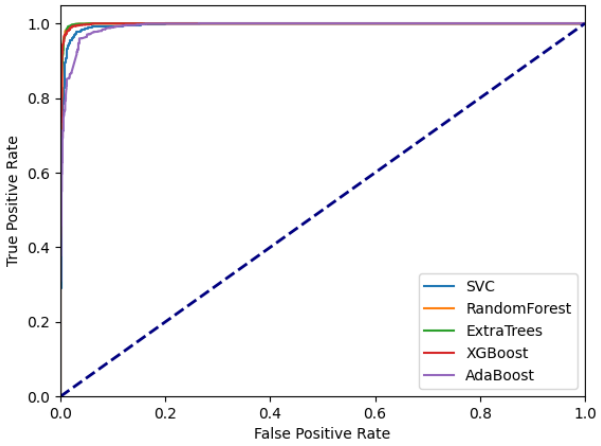

Fig. 2. ROC curve for the Enron dataset

presented in Figure 3. As observed in the figure, across all the datasets, email 'Body' exhibits the foremost importance in all ML algorithms, except in SVM. Surprisingly, SVM considers the 'Sender' attribute, when present, as the most important feature. For the two datasets (i.e., Ling and Enron) having only two features (i.e., 'Subject' and 'Body') SVM considers the 'Subject' as more important than the 'Body'. This must have caused SVM to perform the poorest among the five ML algorithms used in our work.

As seen in Figure 3, the importance of other features varies in different algorithms for each curated dataset. Almost every ML algorithm (except SVM) puts more emphasis on features such as 'Sender' or 'Receiver' or even 'Date' in some cases than the 'Subject' feature. On the contrary, all of the five ML algorithms assign the least importance to the presence or absence of the 'Urls' feature.

TABLE IV
NORMALIZED FEATURE IMPORTANCE SCORES AND RANKS*

Dataset	Best ML Algorithm	Features					
		Sender	Receiver	Date	Subject	Body	Urls
Assasin	ADB	0.06	0.12	0.10	0.04	0.68	0.00
	Rank	4	2	3	5	1	6
TREC-05	ET	0.09	0.08	0.10	0.06	0.61	0.05
	Rank	3	4	2	5	1	6
TREC-06	XGB	0.06	0.03	0.02	0.08	0.80	0.01
	Rank	3	4	5	2	1	6
TREC-07	ET	0.08	0.20	0.02	0.07	0.62	0.01
	Rank	3	2	5	4	1	6
CEAS-08	ET	0.05	0.27	0.01	0.07	0.60	0.00
	Rank	4	2	5	3	1	6
Ling		Subject	Body	Enron		Subject	Body
	XGB	0.00	1.00		ET	0.09	0.91
	Rank	2	1		Rank	2	1

*Here, rank 1 = most important

For a deeper understanding, we focus on the best-performing ML algorithms for each of the seven datasets, because these can give us insights into how these algorithms emphasized certain features to achieve better performances. For each curated dataset, in Table IV, we identify the ML algorithm performing best on the dataset, and we present the normalized importance scores obtained by the algorithm for the available features. For the convenience of interpretation, we also rank the features for the algorithm-dataset pairs. Without surprise, it is observed in Table IV that across all the datasets, the email 'Body' is ranked 1 (i.e., the most important) in all the ML algorithms at their best performance while the 'Urls' feature is consistently regarded as the least important with rank 6. To our surprise, the 'Sender' feature is not given much importance either.

From the observations discussed above, we realize that the simple absence/presence of URLs doesn't contribute much to the ML algorithms' detection of phishing emails. A better URL analysis with an indication of whether or not a URL is malicious can lead to a useful feature. Moreover, a deeper analysis of whether or not the 'Sender' email addresses actually correspond to the claimed legitimate senders (e.g., name, organization) can also lead to an impactful feature for distinguishing phishing emails.

V. THREATS TO VALIDITY

Our phishing email detection mechanism is grounded in the assessment of seven curated datasets we created. These datasets include emails written in English only. Consequently, the broad applicability of our findings may be potentially constrained due to this linguistic preference. Exploring multi-lingual datasets might offer additional insights and challenges in phishing email detection.

In this work, we consider only the *presence* of URLs in the email body. However, a comprehensive analysis of the full URL links, studying their structure, domain, and other attributes, could potentially unearth more intricate patterns and indicators of phishing attempts. The attachments to the emails are not considered in this work. We plan to address this limitation in our future work.

On our curated datasets, we only apply five prominent ML algorithms none of which involved deep learning. This can be argued as a limitation of our work. However, without deep learning, we have achieved 98% or higher accuracy in all the datasets. Moreover, the objective of our work has been to produce curated datasets and identify the features that impact most on the ML algorithms. Nevertheless, we plan to explore deep learning algorithms for the purpose in the future.

VI. RELATED WORK

ML and NLP methods have been utilized in numerous fields [26]–[32], including various security elements of software applications [33]–[36]. Similarly, recent endeavors to identify phishing emails have utilized ML and NLP methods [6], [8], [22], [37]. Some research has focused on using just one algorithm [5], [6], [38], while others have evaluated multiple algorithms to determine the best ones [22], [23]. There are studies that used individual datasets [39] or various datasets [6], [20], [22] and even amalgamations of several datasets [22]. Some looked into feature analysis [7], [8], while some analyzed content of the emails [40]–[42].

Several recent initiatives to improve phishing email detection incorporated invoking suspicion [43], exploring attack

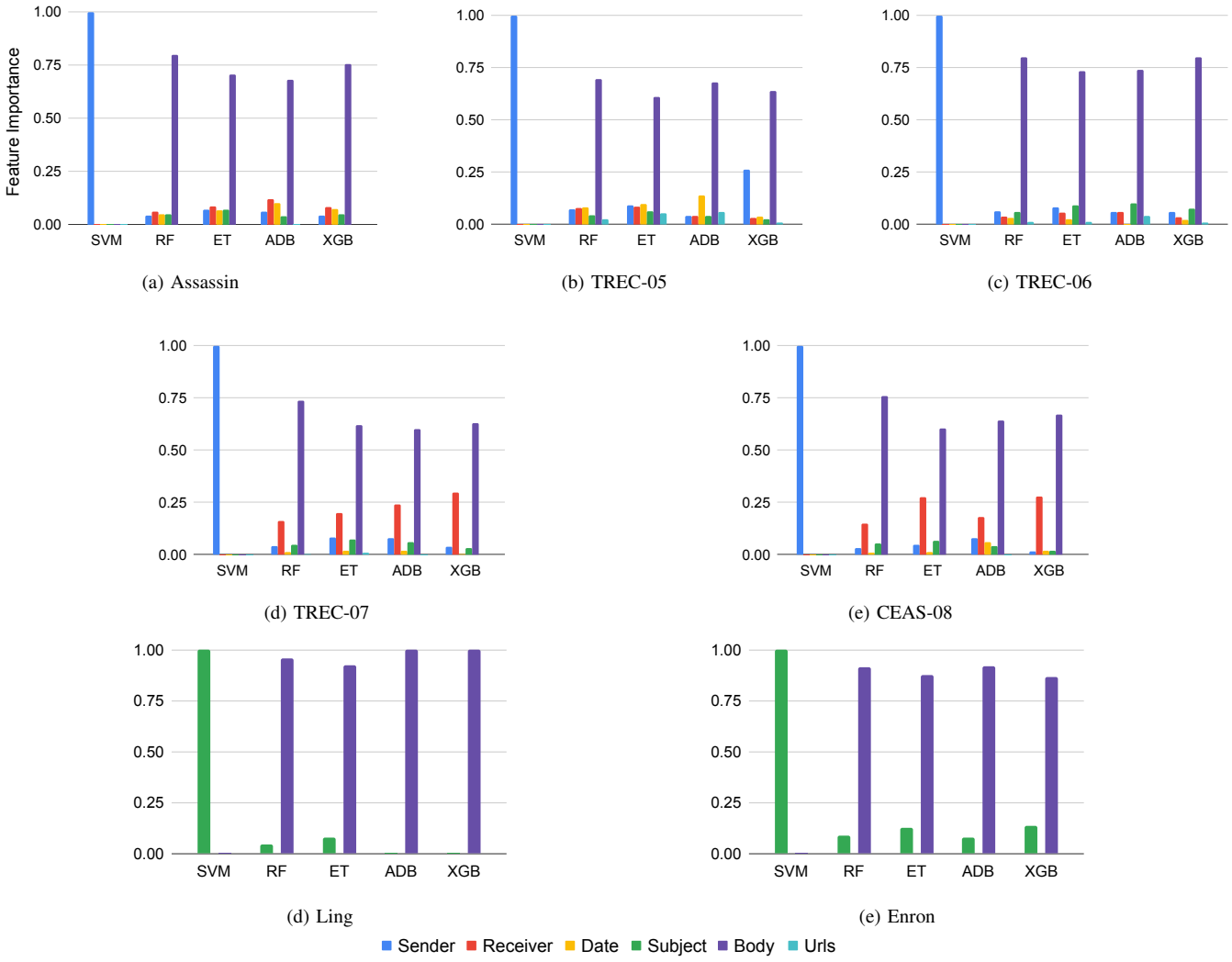

Fig. 3. Relative importance of different features to the five different ML algorithms when operated on our seven curated datasets

tactics and intentions [42]. Das et al. [9] explored the nuances of phishing and spear phishing through unique security challenges. They emphasized the need for rich curated datasets and this work of ours has produced and released seven curated datasets. In addition, our work also includes a feature analysis upon operating five ML algorithms on our curated datasets.

Some studies [7], [8] highlighted dataset features, while some focused on different email languages such as Arabic [21], [37]. Given a typical phishing synopsis, there is a notable disparity in volume between phishing and legitimate emails. A 1:1 ratio is no longer popular among many researchers [44]. In our work, we gauged the efficiency of five ML algorithms on both balanced and imbalanced datasets.

ML techniques such as Naive Bayes, SVM, RF, Decision Tree, Logistic Regression, AdaBoost, XGBoost, and K-Nearest Neighbors are widely used for phishing detection [23], [38], [45]. Agarwal and Kumar [39] fused Naive Bayes with Particle Swarm Optimization and operated on only one dataset. Islam et al. [22] assessed four ML algorithms focusing on specific features of each. Rabbi et al. [20] applied six ML algorithms on only two datasets. In contrast, we operate five prominent ML algorithms on seven curated datasets that we have created. Rabbi et al. [20] reported that the 'subject' feature was key for classification while our findings based on larger datasets report otherwise.

VII. Conclusion

In this work, we first create seven curated datasets by processing repositories of emails from different sources. These datasets contain 203,176 curated emails ranging from the years 2000 through 2008 including personal to professional email communications covering diverse topics of interest. We have released these datasets for free public use [10] and our curated datasets are prepared and ready for application of ML algorithms or similar analyses. The release includes both balanced and imbalanced datasets of varying sizes to offer

varying challenges to the ML algorithms to be operated on them.

We demonstrate the suitability and readiness of our curated dataset by applying five ML algorithms achieving high accuracy in phishing email detection. We also quantitatively examine the significance of different email features in the ML algorithms' decision-making and thus derive insights into why certain algorithms performed better than others and what can be done in the future to enhance the effectiveness of phishing email detection. Our quantitative analysis reveals that some ML algorithms (e.g., SVM) emphasize on wrong features resulting in poor performances. We also reveal that certain features such as URLs and sender information are not being exploited enough at the current state of the art.

We plan to address these by incorporating URL analysis to indicate whether or not a URL is malicious, and sender analysis to indicate whether or not a sender email address legitimately belongs to the claimed person or organization. We plan to conduct a qualitative analysis of misclassified emails to reveal patterns causing ML algorithms to miss phishing emails. In addition, we will further extend our curated datasets and assess their usefulness with deep learning algorithms.

REFERENCES

[1] M. Bach, T. Kamenjarska, and B. Žmuk, "Targets of phishing attacks: The bigger fish to fry," *Procedia Computer Science*, vol. 204, pp. 448–455, 2022.

[2] MimeCast, *How to Stop Phishing Attacks (Whitepaper)*. https://www.mimecast.com/resources/white-papers/how-to-stop-phishing-attacks/, Verified: Sept 2023.

[3] "The latest 2023 phishing statistics (updated august 2023)," 2023.

[4] N. James, "81 phishing attack statistics 2023: The ultimate insight," Verified: Sept 2023.

[5] S. Dhavale, "C-asft: convolutional neural networks-based anti-spam filtering technique," in *Proceeding of ICCSA*, 2020, pp. 49–55.

[6] W. Pan, J. Li, L. Gao, L. Yue, Y. Yang, L. Deng, and C. Deng, "Semantic graph neural network: a conversion from spam email classification to graph classification," *Scientific Programming*, vol. 2022, pp. 1–8, 2022.

[7] M. Alam, D. Sarma, F. Lima, I. Saha, R. Ulfath, and S. Hossain, "Phishing attacks detection using machine learning approach," in *ICSSIT*, 2020, pp. 1173–1179.

[8] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "Phishing email detection using natural language processing techniques: A literature survey," *Procedia Computer Science*, vol. 189, pp. 19–28, 2021.

[9] A. Das, S. Baki, A. El Aassal, R. Verma, and A. Dunbar, "Sok: a comprehensive reexamination of phishing research from the security perspective," *Commun Surv Tutor*, vol. 22, no. 1, pp. 671–708, 2019.

[10] *Seven Phishing Email Datasets*. https://doi.org/10.6084/m9.figshare.25432108.v1, 2024.

[11] A. Champa, M. Rabbi, and M. Zibran, "Curated datasets and feature analysis for phishing email detection with machine learning," in *ICMI*, 2024, pp. 1–7 (to appear).

[12] G. Sakkis, I. Androutsopoulos, G. Paliouras, V. Karkaletsis, C. D. Spyropoulos, and P. Stamatopoulos, "A memory-based approach to anti-spam filtering for mailing lists," *Information retrieval*, vol. 6, pp. 49–73, 2003.

[13] B. Klimt and Y. Yang, "The enron corpus: A new dataset for email classification research," in *ECML*, 2004, pp. 217–226.

[14] A. Schwartz, *Apache SpamAssassin*. https://spamassassin.apache.org/, Verified: August 2023.

[15] N. Craswell, A. De Vries, and I. Soboroff, "Overview of the trec 2005 enterprise track." in *TREC*, vol. 5, 2005, pp. 1–7.

[16] A. Bratko, B. Filipic, and B. Zupan, "Towards practical ppm spam filtering: Experiments for the trec 2006 spam track." in *TREC*, 2006.

[17] G. V. Cormack, "Trec 2007 spam track overview," in *Proc. of TREC*, vol. 500, 2007, p. 274.

[18] D. DeBarr and H. Wechsler, "Spam detection using random boost," *Pattern Recognition Letters*, vol. 33, no. 10, pp. 1237–1244, 2012.

[19] Python. (2023) email — an email and mime handling package. [Online]. Available: https://docs.python.org/3.8/library/email.html

[20] M. Rabbi, A. Champa, and M. Zibran, "Phishy? detecting phishing emails using ml and nlp," in *SERA*, 2023, pp. 77–83.

[21] S. Salloum, T. Gaber, S. Vadera, and K. Shaalan, "A new english/arabic parallel corpus for phishing emails," *TALLIP*, vol. 22, pp. 1–17, 2023.

[22] M. Islam, M. Al Amin, M. Islam, M. Mahbub, M. Showrov, and C. Kaushal, "Spam-detection with comparative analysis and spamming words extractions," in *ICRITO*, 2021, pp. 1–9.

[23] Y. Murti and P. Naveen, "Machine learning algorithms for phishing email detection," *JLISS*, vol. 10, no. 2, pp. 249–261, 2023.

[24] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc., 2022.

[25] Scikit-Learn. (2023) sklearn.ensemble.randomforestclassifier. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[26] M. F. Rabbi, A. I. Champa, M. F. Zibran, and M. R. Islam, "Ai writes, we analyze: The chatgpt python code saga," in *MSR*, 2024, pp. 1–5 (to appear).

[27] A. I. Champa, M. F. Rabbi, M. F. Zibran, and M. R. Islam, "Insights into female contributions in open-source projects," in *MSR*, 2023, pp. 357–361.

[28] M. Zibran, "On the effectiveness of labeled latent dirichlet allocation in automatic bug-report categorization," in *ICSE*, 2016, pp. 713–715.

[29] A. I. Champa, M. F. Rabbi, and M. F. Zibran, "Chatgpt in action: Analyzing its use in software development," in *MSR*, 2024, pp. 1–5 (to appear).

[30] S. M. Mahedy Hasan, M. F. Rabbi, A. I. Champa, and M. A. Zaman, "An effective diabetes prediction system using machine learning techniques," in *ICAICT*, 2020, pp. 23–28.

[31] A. I. Champa, M. F. Rabbi, S. M. Mahedy Hasan, A. Zaman, and M. H. Kabir, "Tree-based classifier for hyperspectral image classification via hybrid technique of feature reduction," in *ICICT4SD*, 2021, pp. 115–119.

[32] M. Islam, M. Ahmmed, and M. Zibran, "Marvalous: Machine learning based detection of emotions in the valence-arousal space in software engineering text," in *ACM SAC*, 2019, pp. 1786–1793.

[33] A. Rajbhandari, M. Zibran, and F. Eishita, "Security versus performance bugs: How bugs are handled in the chromium project," in *SERA*, 2022, pp. 70–76.

[34] A. Champa, M. Rabbi, F. Eishita, and M. Zibran, "Are we aware? an empirical study on the privacy and security awareness of smartphone sensors," in *SERA*, 2023, pp. 1–8.

[35] M. Islam, M. Zibran, and A. Nagpal, "Security vulnerabilities in categories of clones and non-cloned code: An empirical study," in *ESEM*, 2017, pp. 20–29.

[36] M. F. Rabbi, A. I. Champa, C. Nachuma, and M. F. Zibran, "Sbom generation tools under microscope: A focus on the npm ecosystem," in *ACM SAC*, 2024, pp. 1–9 (to appear).

[37] R. Al-Yozbaky and M. Alanezi, "Detection and analyzing phishing emails using nlp techniques," in *HORA*, 2023, pp. 1–6.

[38] R. Ibrahim, M. Argungu, and I. Mungadi, "Development of an ensemble classification model based on hybrid filter-wrapper feature selection for email phishing detection," *International Journal of Computer and Systems Engineering*, vol. 17, no. 9, pp. 519–523, 2023.

[39] K. Agarwal and T. Kumar, "Email spam detection using integrated approach of naïve bayes and particle swarm optimization," in *ICICCS*, 2018, pp. 685–690.

[40] Y. Teo, "Phishing attack detection using machine learning techniques," Ph.D. dissertation, TAR UMT, 2023.

[41] B. Ampel, Y. Gao, J. Hu, S. Samtani, and H. Chen, "Benchmarking the robustness of phishing email detection systems," 2023.

[42] D. Bera, O. Ogbanufe, and D. Kim, "Towards a thematic dimensional framework of online fraud: An exploration of fraudulent email attack tactics and intentions," *Decision Support Systems*, p. 113977, 2023.

[43] M. Harris and D. House, "Invoking suspicion for improved accuracy in phishing email identification," 2023.

[44] M. Zareapoor and K. Seeja, "Feature extraction or feature selection for text classification: A case study on phishing email detection," *IJIEEB*, vol. 7, no. 2, p. 60, 2015.

[45] P. Dinesh, M. Mukesh, B. Navaneethan, R. Sabeenian, M. Paramasivam, and A. Manjunathan, "Identification of phishing attacks using machine learning algorithm," in *E3S Web of Conf.*, vol. 399, 2023, p. 04010.