PHISHING EMAIL CLUSTERING USING QUASI-GROUND-TRUTH MACHINE LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Fast and precise detection of new and unknown phishing campaigns is essential for incident response teams and, more broadly, for overall cybersecurity. However, no reliable automated phishing detection methods exist, largely due to varying incident clustering criteria and the scarcity of ground-truth datasets. We address the problem by providing methods for creating quasi-ground truth, enabling the development of two machine-learning clustering models with a Siamese architecture. The models, trained on a new dataset of phishing email, prove transferable onto a test dataset of a different origin. While the generalization of the model's application is not fully satisfactory, we find that it may be improved through cooperation between the two models. The new phishing dataset has been released publicly.

1 INTRODUCTION

The rapid and accurate detection of phishing campaigns remains essential for incident-response teams and, ultimately, for society at large. Computer emergency response teams (CERTs) operate worldwide in various organizational forms and coordinate their activities to different extents. Nevertheless, as entities responsible for cyber-threat detection, warning, analysis, mitigation, and neutralization, they commonly seek patterns that connect observed incidents. In addition to processing user-reported cases, the proactive collection of naturally observed phishing emails constitutes a valuable mode of CERT operation. The typical use cases for effectively grouping spam emails, and phishing messages in particular, are (i) incident-response triage: security teams must process large volumes of reports rapidly; (ii) attacker or infrastructure profiling: security teams need to identify originating resources (Hamid & Abawajy, 2013); (iii) threat-intelligence enrichment: security teams require deeper campaign insight.

The objective of this practical study is to support CERT operations in addressing these use cases by providing efficient models for phishing email clustering. The reported work was conducted as part of the <anonymized> project, which aims, among other goals, to deliver appropriate tools to the CERT operating at <anonymized>. In this study, we process incoming phishing emails, propose and refine two clustering approaches for different modalities, and evaluate their performance both individually and in tandem. By relying on quasi-ground truth extracted from naturally unlabeled spam data, we address two central challenges in automated spam clustering into campaigns: the definition of a campaign itself and the scarcity of ground-truth data.

1.1 DEFINITIONS OF PHISHING CAMPAIGNS

Phishing email clustering can be understood as a complementary task to phishing detection. Whereas detection asks whether a message is malicious, clustering asks which messages are related and should therefore be interpreted, investigated, or mitigated jointly. This distinction is important because phishing response teams face large streams of user-reported incidents or proactively collected email messages, many of which are variants of the same underlying attack. Althobaiti et al. (2023) and Saka et al. (2022) claim that grouping such messages can support faster triage and more effective application of countermeasures.

The literature does not converge on a single definition of a phishing campaign. A **source-based approach** groups emails according to common source attributes, such as origin, infrastructure, sender

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

artifacts, or IP addresses. A source-based definition is attractive because infrastructure indicators can be relatively objective; however, according to Rashid et al. (2024) it may fail when attackers rotate domains, compromise legitimate services, or use shared phishing kits across unrelated operations. A **narrative-based approach** groups emails according to the deceptive scenario they implement, including the impersonated entity, requested action, lure narrative, and related elements, so that similar cases can be handled together. A **response-based approach** defines a group in terms of the analyst action required for mitigation, such as blocking a domain, removing emails from inboxes, warning users, or updating detection rules.

1.2 Datasets and Ground-Truth Scarcity

The lack of public phishing-email datasets labelled at the campaign level remains a central obstacle in this research area. Most phishing datasets provide binary labels, but do not include clustering information. Semi-supervised clustering methods can make efficient use of a small amount of initially labelled data (Bair, 2013). Realistic campaign clustering often requires organisational mail-report datasets; however, such datasets are difficult to share because emails may contain personal data, internal context, user identifiers, and other proprietary information. As a result, researchers rely on a limited number of real-world reported phishing datasets, which restricts reproducibility and makes systematic comparison of results difficult (Althobaiti et al., 2023; Saka et al., 2022; 2025).

Spam research offers larger-scale examples, but it is not a direct analog for phishing campaign analysis. Anderson et al. (2007) analyse large volumes of spam and identify distinct scams and hosting infrastructure, while Xie et al. (2008) identify botnet-based spam campaigns using large-scale email data. Although these studies demonstrate the value of scale, they also show that labels depend strongly on the observation point and grouping criterion, which may not transfer directly to phishing-report datasets.

In this study, we address both of the above challenges by formally defining a campaign as a set of messages, $C_i = \{m_{ij}\}$. Although the precise rules used by a human operator to group messages are unknown and may vary, we assume that vectors of attributes can be computed for each message, $m_{ij} = (\boldsymbol{u}, \boldsymbol{v})$, such that mathematical models can be trained on these representations and used, either individually or jointly, to cluster messages into disjoint sets that approximate $\mathbb{C}_1, \mathbb{C}_2, \ldots$.

After a review of related works in Sec. 2, we provide the details of the approach and the datasets in Sec. 3, and examine its performance in Sec. 4. While the vector-based email clustering, although done by us in an original way and on a new and proprietary dataset, is an established methodology, we also ask and answer the research questions of (**A**) how well individual Siamese models, trained on a given phishing dataset, **generalize** to other datasets, and (**B**) how effective is the **fusion** of individual models at cluster level. We conclude the work in Sec. 5.

2 Related Work

Following the methodology proposed by Saka et al. (2025), we refer to research, similar w.r.t. clustering algorithms and features of samples on their input.

2.1 Input Features and Representations

Header, Sender and Origin Features. Althobaiti et al. (2023) found that email-origin and URL-based features were especially useful when clustering reported phishing emails into campaigns that are useful for analysts. Similar conclusions about the importance of source address analysis were stated earlier in spam-botnet research by Xie et al. (2008). The advantage of origin features is that they often map to concrete mitigations, such as blocking a sending domain or Internet protocol (IP) address range. The disadvantage is adversarial instability: etc. Thus, origin features are useful but rarely sufficient as a standalone campaign definition.

URL, Domain and Landing-Page Features. URLs in message body are strong phishing indicators because many phishing emails are designed to direct victims to a credential-harvesting page or malware payload.Althobaiti et al. (2023) report that URL-related features, combined with origin features, produced useful clusters for campaign identification. Additionally, prior works by Anderson et al. (2007) find that following spam URLs to destination websites and using such landing-page

similarity can reveal scam infrastructure invisible in raw email text. Beside handling URL shorteners, clustering methods should also handle per-recipient tracking links, compromised legitimate domains, and fast-flux infrastructure. Rashid et al. (2024) work on phishing-URL generalisation using unsupervised domain adaptation goes beyond, addressing the broader distribution-shift problem, i.e. that URL models trained in one domain or period may generalise poorly to new campaigns or collections. For campaign clustering, this implies that URL similarity should be treated as one view among several, not as a universal ground truth.

Textual, Semantic and Contextual Features. Textual similarity remains central to scam-based clustering, starting with traditional representations such as bag-of-words, TF-IDF, n-grams etc. More recent works use semantic embeddings, topic models, and transformer representations to capture paraphrased lures and template variants. Saka et al. (2022) combine contextual and semantic features from phishing emails and compare clustering algorithms over different feature sets. Tiwari (2025) similarly uses sentence-transformers and BERTopic to group phishing texts by semantic patterns and extract interpretable topics. Semantic features are useful when attackers vary URLs, sender names, or superficial wording while preserving the same scam. However, they may over-generalise because many unrelated phishing emails share common persuasive patterns such as urgency, account verification, or reward claims (Tiwari, 2025; Saka et al., 2024).

2.2 Clustering and Grouping Models

Categorical and Rule-Based Clustering. Categorical-feature clustering appears prominently in spam-campaign work. Alishahi et al. (2015) propose Categorical Clustering Tree to cluster spam emails into campaigns using categorical attributes, with the aim of discovering smaller groups, further analyzed according to behavior. Rule-based or criterion-based grouping also appears in operational systems. US patent No. 11895137B2 by Cohen et al. (2024), postulating automated generation of memory-efficient clustered data structures, criteria-based analysis, scoring, and human-readable summaries for phishing data items.

Partitional and Centroid-Based Clustering. Centroid-based methods such as k-means are attractive because of their simplicity and scalability, and they can be combined with TF-IDF, embeddings, or engineered features (Catal et al., 2022; Tiwari, 2025). However, they require the number of clusters to be specified or estimated, which is unknown in a real setting where the number of campaigns changes over time.

Density-Based and Mode-Seeking Clustering. Density-based and mode-seeking models are more aligned with campaign discovery because they do not necessarily require a fixed number of clusters. Althobaiti et al. (2023) find that Mean Shift algorithm outperforms DBSCAN. Saka et al. (2022) also evaluate multiple unsupervised clustering techniques using contextual and semantic feature combinations. DBSCAN can identify noise and discover arbitrary-shaped clusters, which is useful when many reports are singleton or unrelated. Yet it is sensitive to distance metrics, density thresholds, and feature scaling. Mean Shift avoids specifying the number of clusters but depends on bandwidth selection, which may cause over-merging or fragmentation depending on the chosen representation.

Hybrid Clustering-Classification Models. Some recent phishing work combines clustering with classification. Al Shalabi & Jazyah (2024) propose a hybrid phishing-detection method based on clustering and machine-learning classification, combining deep learning and decision-tree components. While this line of work is more detection-oriented than campaign-oriented, it is relevant because clustering can be used as a pre-processing step, representation-learning mechanism, or weak-supervision tool where labeled data are scarce. The limitation is that detection accuracy does not imply campaign-clustering quality. A model may classify phishing emails correctly while failing to distinguish separate campaigns, or it may form clusters that improve classification but are not meaningful to analysts.

Weakly-supervised approaches. Self-training has been investigated for spam filtering by using a classifier trained on labeled email to produce labels or feedback for unlabeled messages, although its effectiveness can deteriorate under distribution shift (Mojdeh & Cormack, 2008). Graph-based semi-supervised spam filtering has also demonstrated that supervision from very few labeled messages can be propagated through the similarity structure of a large email collection (Mojdeh & Cormack, 2010). However, these approaches focus primarily on spam classification rather than us-

ing pseudo-labels to learn campaign-level clusters, leaving pseudo-label-assisted spam campaign clustering comparatively underexplored (Cascante-Bonilla et al., 2021).

3 Materials and Methods

Prior work has demonstrated the feasibility of clustering reported phishing emails, while also highlighting several unresolved challenges, including the scarcity of campaign-labelled datasets, the absence of a consistent definition of a campaign, and the limited evaluation of analyst-oriented utility. Recognising the multi-faceted structure of phishing emails and the different criteria by which phishing campaigns may be defined, we base our approach on two semantically rich components of an email message: its textual content and the URLs embedded within it. We first collect and curate a dedicated dataset, then develop two complementary email representations, together with two corresponding clustering models.

3.1 The spamtrap phishing dataset

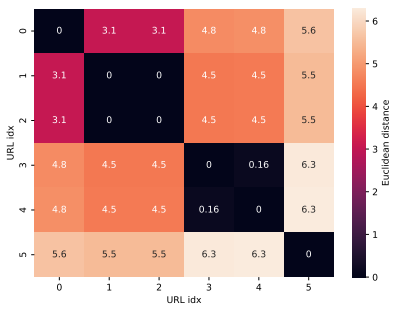
For this study, we used a manually curated subset of phishing-only messages obtained from a proprietary spamtrap infrastructure. Messages were restricted to English using a language detection library Shuyo (2010) ported to Python by Danilák (2021) — and de-duplicated. An email was then preliminarily classified as phishing if it (i) contained any link published in the <anonymized incident database>, or (ii) contained identifiers of the most popular cryptocurrencies, or (iii) originated from an IP address listed by the Spamhaus blocklist service (The Spamhaus Project, 2026). Each of the URL criteria (i)–(iii) can already be considered multidimensional and potentially spanning the present and future attack surfaces.

3.2 Message attributes and quasi-GT development

In our study, the attribute vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ are derived from email textual content and embedded URLs, respectively. Additional representations could be developed from other self-contained modalities, or message components, such as attachments, properties of URL-referenced resources, or remaining email headers. We consider textual content and URLs to be self-contained and partially complementary because campaigns may share semantic similarity at the text level, while also reusing links that point to a common exploitation infrastructure, such as a cryptocurrency wallet, malware distribution endpoint, or credential-harvesting service.

The textual content of each email was cleaned by removing URLs, inline binary content, HTML markup, and other non-textual artefacts. The resulting text was vectorised using the 'bert-base-uncased' language model (Hugging Face, 2026). The email embedding was computed as the mean of the model output over the first 512 tokens, and then compressed by means of a PCA transformation using 50 principal components learned from Enron email corpus by Klimt & Yang (2004), with the same 'bert-base-uncased' representation. This procedure produced the vector $\boldsymbol{u}$, and the selected 50 principal components accounted for more than 70% of the observed variance, which we considered sufficient for the purposes of dimensionality reduction.

Each URL was decomposed into its domain component and the remaining path/query component, with each part represented by features matched to its statistical character. The domain component was vectorized using a FastText model (Bojanowski et al., 2017) trained to predict domain-name tokens, producing a 100-dimensional embedding. Because FastText represents words as bags of character n-grams, this embedding captures sub-lexical structure, so domains sharing morphological patterns (e.g., URL shortener or similarly generated domains) are embedded close together even when they share no exact tokens. In turn, the process of path/query string generation, results in a peculiar symbol distribution, making Shannon (1948) entropy a stable and discriminative signature. Consequently, the path component was split into segments and the Shannon entropy of each segment was individually computed; the first ten elements were appended to FastText embedding, yielding the URL representation $\boldsymbol{v}$ of dimensionality 110. It was verified empirically that paths originating from the same shortener or content generator cluster tightly in entropy space, even when their pairwise edit distances are large, cf. Fig. 1. As shown on the heatmap, URLs from the same shortener exhibit small distances despite their path components being entirely different; a similar observation holds for URLs pointing to the same hosting service. At the same time, distances between



0: https://vo.la/Zh4fQK9;
1: https://t.co/ZW8FPmoAKt;
2: https://t.co/AW9FrvoDst;
3: https://cloudstorageservices1.storage.googleapis.com/31e2c63f-f643-48e3-87f3-1ba6ed97ffb1.html;
4: https://cloudstorageservices1.storage.googleapis.com/47d2f4e1-2dda-4a3b-938b-2ccef20b5385.html;
5: https://www.salomon.com/product/s-lab-ultra-12-lc14298/LC2863300

Figure 1: Pairwise Euclidean distance computed between exemplary URLs

unrelated URLs remain high. Although richer URL representations have been proposed, such as character-level recurrent architectures by Bahnsen et al. (2017), and deep URL-specific encoders by Le et al. (2018), these require training on large labeled URL corpora. Our representation is training-free beyond the unsupervised token-prediction objective and captures the two components of a URL through features matched to their statistical properties.

Because ground truth is generally unavailable in fast, automated email-clustering workflows, we propose a procedure for constructing **quasi-ground truth**, hereafter referred to as quasi-GT. Quasi-GT is intended to approximate the clustering decisions that might be made by a human analyst. We adopt a reciprocal strategy: the quasi-GT used for developing the text-based clustering model is derived from URL similarity, whereas the quasi-GT used for URL-based clustering is derived from textual similarity. We denote these two alternative partitions of the same email corpus by $\mathbb{C}_1^{\boldsymbol{u}}, \ldots, \mathbb{C}_k^{\boldsymbol{u}}$ and $\mathbb{C}_1^{\boldsymbol{v}}, \ldots, \mathbb{C}_k^{\boldsymbol{v}}$, where k denotes the number of reference clusters.

The $\boldsymbol{u}$-related quasi-GT is obtained by applying k-means clustering to the URL-based feature vectors $\boldsymbol{v}$. The construction of the $\boldsymbol{v}$-related quasi-GT is not symmetric: it is based on pairwise textual comparison of all messages, followed by grouping using the Smith–Waterman algorithm Smith & Waterman (1981). The complete workflow, including references to the methods described above, is presented in Fig. 2.

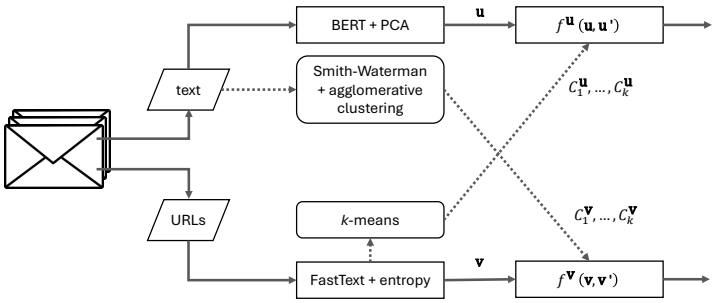


Figure 2: Data flow and processing diagrams. Solid lines denote paths in inference and training phases; dotted lines refer to quasi-GT preparation for the training phase. The workflow assumes that the processed email stream already contains phishing messages only, as a result of pre-classification by one of the established phishing-email detection methods (Fette et al., 2007; Bergholz et al., 2010).

3.3 SIAMESE MODELS

Owing to the open-ended nature of the clustering task and the limited availability of training data, we adopt a Siamese architecture for both classification models, namely the text-based and URL-based models. Siamese networks are particularly suitable for similarity-learning problems in which the set of categories is large, only partially observed, or not fixed during training Chopra et al. (2005). We denote the operation of the two models by the functions $f^{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{u}')$ and $f^{\boldsymbol{v}}(\boldsymbol{v}, \boldsymbol{v}')$, respectively, as illustrated in Fig. 2. The text-processing model consists of the following layers and training

components: (i) input layer accepting feature vectors of length 50; (ii) a fully connected layer with 32 ReLU neurons; (iii) a fully connected layer with 16 ReLU neurons; (iv) a Siamese embedding layer with 8 neurons; (v) L2 unit-length normalization, and (vi) the Euclidean similarity measure.

The proposed URL-processing model adopts an architecture consisting of two identical branches with shared weights. Each branch processes one of the input samples through two fully-connected layers with ReLU activations Agarap (2018), producing a latent representation of size 256. The resulting embeddings from both branches are subsequently concatenated to form a joint feature vector of size 512. This combined representation is passed through two additional fully-connected layers, which yield a binary output $\hat{y} \in [0, 1]$, where $\hat{y} = 1$ indicates that the input samples are predicted to belong to the same cluster, and $\hat{y} = 0$ corresponds to samples assigned to different clusters. Both models were trained using a uniform procedure. The corresponding quasi-GT datasets were divided into a validation subset, consisting of four classes, and a training subset containing the remaining data. Pairs of samples were generated uniformly at the class level.

The proposed quasi-GT construction inevitably introduces a degree of alignment between the two models, since the quasi-GT for one modality is derived from the other modality. The quasi-GT construction procedures differ from the raw vectors $\boldsymbol{u}$ and $\boldsymbol{v}$ provided as model inputs, avoiding a direct tautological dependency. To the best of our knowledge, this is the first attempt to employ an automatically generated substitute for GT in order to enable further machine-learning-based development for phishing-campaign clustering, a task for which labelled data are particularly scarce.

3.4 Evaluation Strategies

Evaluation remains one of the weakest points in phishing-campaign clustering. Internal clustering metrics such as silhouette score can measure compactness and separation, but they do not tell whether a cluster corresponds to a useful campaign. Althobaiti et al. (2023) use silhouette coefficient and homogeneity score, but also conduct a user study to validate such results.

Our work focuses first on performance evaluation based on quasi-GT-labelled own spamtrap dataset, and next, on manually-labelled independent reference dataset provided by Saka et al. (2022). The initial and most direct evaluation approach, naturally suited to Siamese models, is to assess classification accuracy on pairs of test messages. For this purpose, we compute the area under the receiver operating characteristic curve (ROC AUC) and the equal error rate, defined as the error at the intersection of the false-acceptance-rate and false-rejection-rate curves. These measures allow model performance to be evaluated across varying classification thresholds. Simultaneously, we compare results of our Siamese-based approach to the one proposed by Saka et al. (2022), based on concatenated embeddings for message text and subject. Next, to assess structural clustering properties, the clusters produced by our models are compared against clusters provided mutually or externally, with Hungarian algorithm (Kuhn, 1955; Munkres, 1957), and normalized to the interval $[0, 1]$.

Clusters can be reconstructed from Siamese models in several ways. For models that directly compare the vector outputs of the two branches, embeddings for all test samples can be computed and subsequently clustered using the k-means algorithm. For Siamese models, the pairwise-distance information can be used to perform k-medoids clustering Kaufman & Rousseeuw (1990) or agglomerative clustering. All of the above evaluation approaches have been applied in our study.

3.5 Clustering based on joint operation of the two Siamese models

Although the two Siamese models $f^{\boldsymbol{u}}(\boldsymbol{u}, \boldsymbol{u}')$ and $f^{\boldsymbol{v}}(\boldsymbol{v}, \boldsymbol{v}')$ can be used independently for clustering, we assume that they capture partially complementary information. To exploit this complementarity, we propose their joint application. However, because of the structural complexity of the URL-based model head, a simple concatenation of branch-output vectors from the two models is unlikely to be effective. Instead, we aggregate the models at the level of pairwise distance computation. Specifically, the distance produced by the text-based model is assigned weight α, whereas the distance produced by the URL-based model is assigned weight $1 - \alpha$. Aggregation at the distance level can be done by sum, minimum or maximum operators, enabling subsequent clustering as above.

4 RESULTS

Evaluation of the developed methods was first carried out on quasi-GT derived from the spamtrap dataset, mostly to verify the correctness and usability of the selected approaches. Next, proper testing was performed on the reference dataset, which was created and labeled under remarkably different conditions. This part was divided into verification of individual Siamese model operation and into experiments on the joint application of the models.

4.1 SIAMESE MODEL PERFORMANCE FOR QUASI-GT TRAINING

Both Siamese models were trained multiple times using randomly selected validation classes. The text-based model training parameters were: learning rate 10^{-3}; number of epochs was variable, capped at 100; and early stopping parameters: monitoring validation loss, with a patience of 5, restoring the best weights. The URL-based model was trained for 300 epochs using PyTorch (Paszke et al., 2019) with a learning rate of 10^{-3}. Alternative configurations, including lower learning rates and L2 regularization via weight decay, were evaluated but did not yield any performance improvements. The number of training epochs was determined empirically by monitoring the training loss and validation ROC AUC. This setting was sufficient to achieve stable performance, with no observable signs of overfitting on the validation set. For both models, AUC varies from 0.7 to 0.99 and EER varies from 0.05 to 0.3, which is results from randomness in validation class selection and mark models' sensitivity to data. However, most of the samples are gathered close to 1.0 AUC and 15% EER — and can be considered acceptable.

These results from our approach, provided synthetically by figures in the 1st row in Tab. 1, have been compared against the approach by Saka et al. (2022), provided in the row below, and proved superior. This confirms that Siamese model can not only make up for the loss of 30% of the explained variance, but also really learn some mapping pattern between the modalities.

4.2 INDIVIDUAL MODELS TRANSFERABILITY

Table 1: Comparison of Evaluation Results

Evaluation		Model type: text-only		url-only		joint	
		result					
dataset	method	best	mean	best	mean	best	mean
quasi-GT	pairwise AUC	0.98	0.95	0.99	0.90	—	
	pairwise AUC*	0.86	0.86	—-	—	—	
reference†	pairwise AUC	0.81	0.75	0.71	0.61	—	
	k-means fit	0.59	0.48	—		—	
	k-medoids fit	—		—		0.67	0.55
	agglomerative fit	—		0.61	0.51	0.70	0.61

*Vectorization method as proposed by Saka et al. (2022), run once for the whole spamtrap dataset

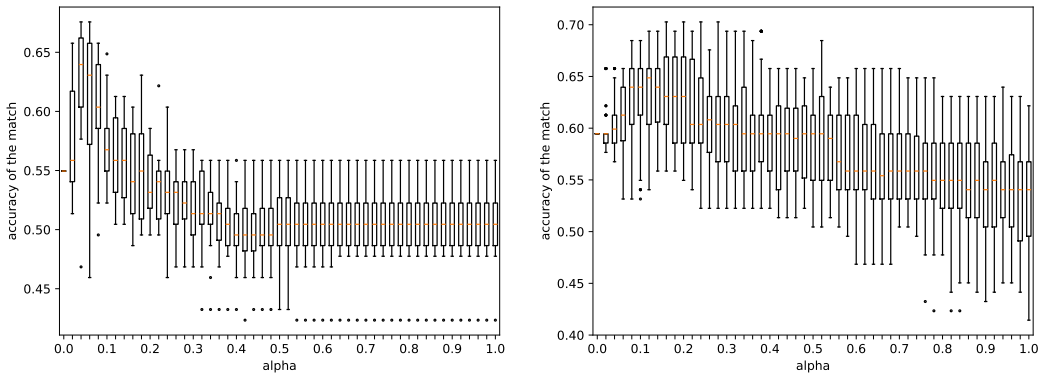
†Mail dataset as provided by Saka et al. (2022)

Models' transferability to a different dataset was evaluated using the same evaluation scheme: messages were compared pairwise, maintaining an equal ratio of same-class to different-class pairs. As the result, AUC drops by 20 to 30 percentage points on the reference dataset, providing a quantitative estimate of transferability. Consistently, EER rises roughly by the same amount — yet it still demonstrates that both models' operation is far from random (cf. figures in 3rd row of Tab. 1).

We have also checked the structural performance metrics, as outlined in Sec. 3.4. Text-based model output was clustered with k-means, while URL-based one — with agglomerative algorithm (standard Python implementation in package `scikit-learn`, linkage type: `complete`). The match score with original clusters, calculated with the Hungarian algorithm, is provided in Tab. 1. Evaluation of result variability was performed by running a number of models trained on quasi-GT on the reference dataset. Experiments showed little difference (2–3 p.p.) between the models performance in subsequent runs.

Note that to make the URL-based model applicable, we removed URL-missing mail records from the reference dataset. Also, we removed clusters with a cardinality of less than 2. Eventually, the modified reference dataset contains 111 records grouped in 6 classes (campaigns).

4.3 JOINT MODEL OPERATION



(a) (b)

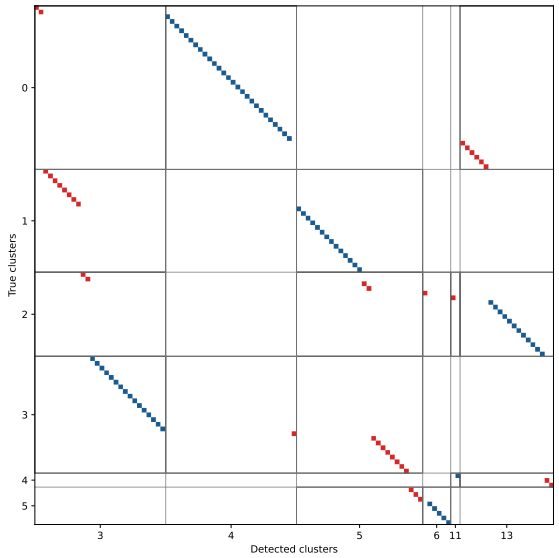
Figure 3: Performance of joint operation of a) k-medoid and b) agglomerative clustering on reference dataset, w.r.t. the ratio of pairwise distances summed from respective models (0.0 — text model only; 1.0 — URL model only), with standard box plot metrics, as in Python `matplotlib` package implementation of the `boxplot` routine.

The experimental verification of the claim that the two models combined provide more useful classification information was done using the scheme presented in Sec. 3.5. For k-medoids, stability of the solutions was tested w.r.t. randomly chosen training/validation subsets of URL pairwise distances (15 draws), times varying algorithm seed (10 draws). For agglomerative clustering, only distance draws counted — but we checked also method operation for different metrics (`L1/euclidean`) and linkage (`ward/average`). Additionally, we verified impact of standard initial distance transformations: $\log_{10}$, median normalization and raising to the power of two. Performance evaluation was done using: (i) best mean match between clusters, w.r.t. α, and (ii) match variability, defined as mean relative change of the match, between adjacent values of α. Comprehensive assessment of Pareto front has led to two prospective configurations, whose operation is shown in Fig. 3a) and b). These are: a) k-medoids with original distances compressed logarithmically, and then normalized to their median, and b) agglomerative clustering on raw distances, with `ward` and `euclidean` options — respectively. In both cases, the distance aggregation was a simple weighted sum. Both model mixtures, while being evidently quite fluctuating in terms of performance, can consistently achieve at $\alpha \approx 0.1$ performance that is better than for any of the composite models used alone. The improvement can also be treated as a proxy measure of efficiency of the quasi-GT approach.

Fig. 4 visualizes the best group mapping details for agglomerative clustering case ($\alpha = 0.06$, draw No. 5). We chose to inspect the best result in detail to see what constitutes the hardest obstacle for the algorithm. In particular, detected cluster #3 (1st column block) matches the ground truth poorly. Next to the matrix, we provide the content of 'Subject' field of the wrongly matched records — only to reveal that all of them refer to the same type of phishing campaign referring to PayPal service (message body is not shown due to the lack of space). Therefore, the mismatch (and drop in accuracy) is rather apparent one, and is due to superficial grouping approach taken in Saka et al. (2022), while the one proposed by us tends to cluster email by its intent, not the formulation.

5 CONCLUSION

This study makes several practical contributions to the field of phishing email clustering: (i) it demonstrates that an appropriately set up spamtrap infrastructure can collect phishing data of sufficient relevance and volume for machine-learning applications; (ii) it introduces an original procedure for constructing a substitute for unavailable ground-truth labels and applies this procedure to obtain



(a)

»WARNING-IFRAME-Unauthorized access to your Pay...
»Notification from Billing Department
»Unauthorized access to your PayPal account!
»FPA NOTICE: Suspicious Activity -Section 9
»FPA NOTICE: Suspicious Activity -Section 9
»IMPORTANT PayPal Credit Union Account Verifica...
»PayPal Account Security Measures
»Update Your Account

(b)

Figure 4: a) Details of the mapping of the agglomerative clustering for weighted pairwise distances. Rows and columns correspond to messages and are grouped horizontally according to the clustering results, and vertically according to GT labels. Dots represent individual correct (blue) and incorrect (red) mappings. b) Message subject for incorrect mappings in 1st group of columns.

a functional dataset suitable for model training; (iii) it develops two lightweight Siamese models for email-message clustering, based respectively on textual and URL-derived representations; and (iv) it evaluates the proposed models on an external reference dataset under multiple clustering configurations.

The presented methodology constitutes a viable and promising approach to automated phishing email clustering and is suitable for application by CERTs and similar operational security teams. Moreover, the proposed models can be retrained incrementally as additional data become available. The approach can also be extended in a straightforward manner by incorporating additional email modalities or, at the clustering stage, temporal information, which may further improve performance.

The experimental results indicate that transferring the models to an independent dataset leads to a performance decrease of approximately 20 percentage points. At the same time, the joint operation of the two models through weighted pairwise distances improves performance by 5–10 percentage points. These findings address research questions **A** and **B**.

A natural question is whether the textual and URL modalities could be exploited more effectively by an integrated multimodal representation. We retain late fusion for two reasons. First, the two modalities may not always be uniformly available: phishing or spam emails can be deliberately crafted in unusual formats to evade detection. In such cases, late fusion degrades gracefully when a modality is absent, whereas a joint encoder presupposes well-aligned inputs in both channels. Second, learning a joint multimodal representation from paired text-URL samples requires labeled training dataset with both modalities (Baltrušaitis et al., 2019), which would reintroduce the supervision this pipeline was designed to avoid.

The text models used in this study cannot fully preserve neither email text nor URL semantics, which is one of the limitations of our method. Furthermore, the Siamese architecture does not inherently adapt to fundamentally new phishing-engineering strategies. Autoencoder-based models may offer greater flexibility in this respect, although at much higher cost and a different deployment profile.

The following two future works appear particularly promising: (i) to improve calibration and preprocessing of URL embeddings, and (ii) to unify classification heads of the two Siamese models, so that model fusion could be performed directly at the level of branch-output representations.

REPRODUCIBILITY STATEMENT

The two datasets are publicly available in anonymized form at < now here, to be replaced finally with Zenodo link>. The implementation details are subject to the project's disclosure policy and may be made available upon request, depending on the scope of the request.

AI USE STATEMENT

In this work, we used generative AI tools for: implementing selected methods. We have not used generative AI tools for: help in developing theoretical models or conceptual frameworks, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, cleaning and reformatting dataset, supporting qualitative and thematic data analysis, interpreting results
— and 'generate synthetic data sets, formulate mathematical claims, provide critical ingredients for proving mathematical claims, assist in the writing of proofs' are not applicable to this work.

Additionally, we used generative AI tools for creating a scientific figure, analyzing existing literature, editing a research paper to improve readability. We have reviewed all AI-assisted work, by manual revision of the code, verification of references, continual reviewing, correction and completion of AI-generated parts of the article. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ACKNOWLEDGMENTS

The research presented in this article was conducted as part of <anonymized> grant whose primary goal is to provide a comprehensive set of AI-based tools to support the critical services of operational teams responsible for cyber defense.
The authors would like to thank <anonymized> team for their substantive assistance and for providing some of the data necessary to conduct the research.

REFERENCES

Abien M Fred Agarap. Deep learning using Rectified Linear Units (ReLU). 2018. URL https://arxiv.org/abs/1803.08375v2.

Luai Al Shalabi and Yahia Hasan Jazyah. Phishing detection using clustering and machine learning. IAES International Journal of Artificial Intelligence, 13(4):4526–4536, 2024. doi: 10.11591/ijai.v13.i4.pp4526-4536.

Mina Sheikh Alishahi, Mohamed Mejri, and Nadia Tawbi. Clustering spam emails into campaigns. In Proceedings of the 1st International Conference on Information Systems Security and Privacy, ICISSP, pp. 90–97. SciTePress, 2015. doi: 10.5220/0005244500900097.

Kholoud Althobaiti, Maria K. Wolters, Nawal Alsufyani, and Kami Vaniea. Using clustering algorithms to automatically identify phishing campaigns. IEEE Access, 11:96502–96513, 2023. doi: 10.1109/ACCESS.2023.3310810.

David S. Anderson, Chris Fleizach, Stefan Savage, and Geoffrey M. Voelker. Spamscatter: Characterizing internet scam hosting infrastructure. In Proceedings of the 16th USENIX Security Symposium, USENIX Security. USENIX Association, 2007.

Alejandro Correa Bahnsen, Dayen Viloria Dibo, Djamila Aouada, and Björn Ottersten. Classifying phishing URLs using recurrent neural networks. In APWG Symposium on Electronic Crime Research (eCrime), 2017.

Eric Bair. Semi-supervised clustering methods. WIREs Computational Statistics, 5(5):349–361, 2013. doi: 10.1002/wics.1270. URL https://doi.org/10.1002/wics.1270.

Tadas Baltrušaitis, Chaitanya Ahuja, and Louis-Philippe Morency. Multimodal machine learning: A survey and taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence, 41(2):423–443, 2019.

André Bergholz, Jan De Beer, Sebastian Glahn, Marie-Francine Moens, Gerhard Paaß, and Siehyun Strobel. New Filtering Approaches for Phishing Email. Journal of Computer Security, 18(1): 7–35, 2010. doi: 10.3233/JCS-2010-0371.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. Transactions of the association for computational linguistics, 5:135–146, 2017.

Paola Cascante-Bonilla, Fuwen Tan, Yanjun Qi, and Vicente Ordonez. Curriculum labeling: Revisiting pseudo-labeling for semi-supervised learning. Proceedings of the AAAI Conference on Artificial Intelligence, 35(8):6912–6920, 2021. doi: 10.1609/aaai.v35i8.16852. URL https://doi.org/10.1609/aaai.v35i8.16852.

Cagatay Catal, Görkem Giray, Bedir Tekinerdogan, Sandeep Kumar, and Suyash Shukla. Applications of deep learning for phishing detection: A systematic literature review. Knowledge and Information Systems, 64:1457–1500, 2022. doi: 10.1007/s10115-022-01672-x.

Sumit Chopra, Raia Hadsell, and Yann LeCun. Learning a Similarity Metric Discriminatively, with Application to Face Verification. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, pp. 539–546. IEEE Computer Society, 2005. doi: 10.1109/CVPR.2005.202.

David Cohen, Jason Ma, Bing Jie Fu, Ilya Nepomnyashchiy, Steven Berler, Alex Smaliy, Jack Grossman, James Thompson, Julia Boortz, Matthew Sprague, Parvathy Menon, Michael Kross, Michael Harris, and Adam Borochoff. Phishing data item clustering and analysis, 2024. URL https://patents.google.com/patent/US11895137B2/en. U.S. Patent US11895137B2, issued February 6, 2024, assigned to Palantir Technologies Inc.

Michal Danilák. Language detection library for python, 2021. URL https://github.com/Mimino666/langdetect.

Ian Fette, Norman Sadeh, and Anthony Tomasic. Learning to Detect Phishing Emails. In Proceedings of the 16th International Conference on World Wide Web, WWW '07, pp. 649–656, New York, NY, USA, 2007. ACM. doi: 10.1145/1242572.1242660.

Isredza Rahmi A. Hamid and Jemal H. Abawajy. Profiling phishing email based on clustering approach. In 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom, pp. 628–635. IEEE, 2013. doi: 10.1109/TrustCom.2013.77.

Hugging Face. google-bert/bert-base-uncased. https://huggingface.co/google-bert/bert-base-uncased, 2026. Accessed: 2026-06-06.

Leonard Kaufman and Peter J. Rousseeuw. Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York, 1990. doi: 10.1002/9780470316801.

Bryan Klimt and Yiming Yang. The Enron Corpus: A New Dataset for Email Classification Research. In Machine Learning: ECML 2004, volume 3201 of Lecture Notes in Computer Science, pp. 217–226, Berlin, Heidelberg, 2004. Springer. doi: 10.1007/978-3-540-30115-8_22.

Harold W. Kuhn. The Hungarian Method for the Assignment Problem. Naval Research Logistics Quarterly, 2(1–2):83–97, 1955. doi: 10.1002/nav.3800020109.

Hong Le, Quang Pham, Doyen Sahoo, and Steven C. H. Hoi. URLNet: Learning a url representation with deep neural networks. In Proceedings of the International AAAI Conference on Web and Social Media (ICWSM), 2018.

Mona Mojdeh and Gordon V. Cormack. Semi-supervised spam filtering: Does it work? In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, pp. 745–746, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 978-1-60558-164-4. doi: 10.1145/1390334.1390482. URL https://doi.org/10.1145/1390334.1390482.

Mona Mojdeh and Gordon V. Cormack. Semi-supervised spam filtering using aggressive consistency learning. In Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, pp. 751–752, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 978-1-4503-0153-4. doi: 10.1145/1835449.1835598. URL https://doi.org/10.1145/1835449.1835598.

James Munkres. Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics, 5(1):32–38, 1957. doi: 10.1137/0105003.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. Advances in Neural Information Processing Systems, 32, 12 2019. ISSN 10495258. URL https://arxiv.org/abs/1912.01703v1.

Fariza Rashid, Ben Doyle, Soyeon Caren Han, and Suranga Seneviratne. Phishing URL detection generalisation using unsupervised domain adaptation. Computer Networks, 245:110398, 2024. doi: 10.1016/j.comnet.2024.110398.

Tarini Saka, Kami Vaniea, and Nadin Kökciyan. Context-based clustering to mitigate phishing attacks. In Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security, AISec '22, pp. 1–12, New York, NY, USA, 2022. ACM. doi: 10.1145/3560830.3563728.

Tarini Saka, Rachiyta Jain, Kami Vaniea, and Nadin Kökciyan. Phishing Codebook: A structured framework for the characterization of phishing emails, 2024. URL https://arxiv.org/abs/2408.08967.

Tarini Saka, Kami Vaniea, and Nadin Kökciyan. Sok: Grouping spam and phishing email threats for smarter security. IEEE Access, 13:54938–54959, 2025. doi: 10.1109/ACCESS.2025.3555157.

Claude Elwood Shannon. A mathematical theory of communication. The Bell System Technical Journal, 27(3):379–423, 1948.

Nakatani Shuyo. Language detection library for java, 2010. URL https://github.com/shuyo/language-detection.

T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. Journal of Molecular Biology, 147(1):195–197, 1981. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(81)90087-5. URL https://www.sciencedirect.com/science/article/pii/0022283681900875.

The Spamhaus Project. Spamhaus Blocklists: Trusted, actionable DNSBLs for email filtering. https://www.spamhaus.org/blocklists/, 2026. Accessed: 2026-06-06.

Himanshu Tiwari. Clustering and topic modeling of phishing texts: A multi-visualization approach, 2025. Preprint.

Yinglian Xie, Fang Yu, Kannan Achan, Rina Panigrahy, Geoff Hulten, and Ivan Osipkov. Spamming botnets: Signatures and characteristics. In Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication, SIGCOMM '08, pp. 171–182, New York, NY, USA, 2008. ACM. doi: 10.1145/1402958.1402979.