



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Using Clustering Algorithms to Automatically Identify Phishing Campaigns

Citation for published version:

Althobaiti, K, Vaniea, K, Wolters, MK & Alsufyani, N 2023, 'Using Clustering Algorithms to Automatically Identify Phishing Campaigns', *IEEE Access*, vol. 11, 10235951, pp. 96502-96513.
<https://doi.org/10.1109/ACCESS.2023.3310810>

Digital Object Identifier (DOI):

[10.1109/ACCESS.2023.3310810](https://doi.org/10.1109/ACCESS.2023.3310810)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Publisher's PDF, also known as Version of record

Published In:

IEEE Access

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



RESEARCH ARTICLE

Using Clustering Algorithms to Automatically Identify Phishing Campaigns

KHOLOUD ALTHOBAITI¹, MARIA K. WOLTERS², NAWAL ALSUFYANI¹, AND KAMI VANIEA²

¹Department of Computer Science, Taif University, Taif 26571, Saudi Arabia

²School of Informatics, The University of Edinburgh, EH8 9AB Edinburgh, U.K.

Corresponding author: Kholoud Althobaiti (kholod.k@tu.edu.sa)

This work was supported by the Deanship of Scientific Research, Taif University.

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT Attackers attempt to create successful phishing campaigns by sending out trustworthy-looking emails with a range of variations, such as adding the recipient name in the subject line or changing URLs in email body. These tactics are used to bypass filters and make it difficult for the information system teams to block all emails even when they are aware of an ongoing attack. Little is done about grouping emails into campaigns with the goal of better supporting staff who mitigate phishing using reported phishing. This paper explores the feasibility of using clustering algorithms to group emails into campaigns that IT staff would interpret as being similar. First, we applied Meanshift and DBSCAN algorithms with seven feature sets. Then, we evaluated the solutions with the Silhouette coefficient and homogeneity score and find that Mean Shift outperforms DBSCAN with email origin and URLs based features. We then run a user study to validate our clustering solution and find that clustering is a promising approach for campaign identification.

INDEX TERMS Phishing, incident response handling, phishing campaign, email clustering, phishing clustering, clustering.


I. INTRODUCTION

Fraudulent emails aim to deceive employees and customers into clicking on malicious links, downloading malware, exposing sensitive information, or transferring money to the attacker, all of which can put organisations and their customers at risk. Supporting and teaching end-users to identify and report phishing emails has been extensively researched; however, detecting the phishing is only the first step to actively protect others. It is necessary that Information Technology (IT) staff can quickly learn about the phishing emails and take actions such as updating blacklists, deleting emails from inboxes, and managing compromised accounts [1], [2]. In order to succeed to apply remediations, IT staff need to process large numbers of reported phishing quickly [1], [3] to find the ones that are new or cases where implemented mitigations are not working. However, this procedure is not

easy because typically, a large number of reports come in, and these need to be handled quickly before end users click links or otherwise engage with the phishing email.

To increase the chance of successfully compromising many people, attackers rely on campaigns where they send several standard phishing emails to random people or semi-targeted phishing to a group of users who are similar in some way, such as working for the same organisation. To evade being blocked by spam filters, attackers may send several versions of the same email, such as creating a unique subject for every recipient or sending the email from several different email servers. The tactic is used to maximise the odds that some of the emails may make it through the automated filters and make it harder for IT defenders to find and delete all the phishing emails [4].

Organisation IT teams rely on users to report phishing that makes it through filters and into inboxes. These reported phish are then used to update the automated filters and remove the phish from inboxes using specific parameters such as date

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque .

received, sender, and email subject [1]. However, because the incoming phish have variations, it is necessary to consider all the reported phish to ensure that the revised filters cover all variations. The problem is that reviewing reports is a manual activity that is time expensive and multiple campaigns can be happening at the same time. So in practice only a few sample reports are used when refining filter rules [1]. While this tactic saves time, it also means that some phishing variations are reported but not accounted for in revised protections and therefore users are not fully protected from that specific campaign. To make full use of all reported phishing, IT teams need an efficient way of grouping reported phish so they can easily consider all similar reports when improving protections.

In this paper, we investigate whether it is feasible to group potential phishing emails into meaningful clusters. Such clusters might then help IT teams analyse potential phishing emails in terms of campaigns, and reduce the effort spent on manually checking each email individually. A successful cluster-based system should result in largely homogeneous clusters where all phishing emails in the same cluster belong to the same phishing campaign and where the number of clusters generated is substantially lower than the original number of individual emails. To this end, we compared seven feature sets and two clustering algorithms, Mean Shift [5] and DBSCAN [6]. Both algorithms are robust to outliers and sparse data.

We applied the algorithms, to two slices of $\approx 60\text{K}$ emails taken from a significantly larger dataset of reported phishing provided by a security services company. Clustering solutions were assessed using an internal validation metrics, Silhouette and a homogeneity metric based on 10 manually identified campaigns in the dataset. We found that the Mean Shift algorithm with the email origin and URL features can reduce the original set of 60K emails to $<6\text{K}$ relatively homogeneous clusters, which are potentially much faster for the IT teams to check.

We also conducted an online study to examine whether the clustering solution of Mean Shift with the email origin and URL features can work well in an organisational environment. The study aims at understanding whether users agree with the clustering or not by showing them a pair of emails and asking them whether they belong to the same cluster or not. We presented 60 pairs of emails from the same clusters and another 60 pairs of emails from different clusters to the participants and found that clustering results are likely accurate when there is a common feature between pairs of emails such as the same email address, email subject or even email date. Similarly, if emails from different clusters have at most one similar value of the aforementioned features, they are unlikely to be marked as being from the same campaign. We discuss how our findings can be used to design a tool that might allow IT staff automatically identify and act on phishing campaigns.

The rest of this paper is structured as follows. First, we summarise previous work on organisational handling of

phishing emails and phishing clustering (Section II). We then describe the dataset and rationale, design, and methodology of our study (Sections III and IV). Our results are organised based on the internal and external validity of the algorithms and feature sets (Section V) and the experts' evaluation of the selected parameters (section VI). We discuss the implications of this result for further work on detecting phishing campaigns in Section VII and conclude with limitation, a plan of future work and summary of the work in Sections VIII and IX.

II. BACKGROUND

A. ORGANIZATIONAL PHISHING MANAGEMENT

Phishing is one of the most common and most disruptive types of attacks organisations face [7], [8]. It is also a gateway attack from which an attacker gains basic level access by tricking staff into revealing their login credentials, and then uses those credentials to launch another more damaging attack [8]. So even if the initial compromise seems small, the final impact on an organisation can be quite large. Consequently, organisations take phishing very seriously and use a range of methods to mitigate its impact, with one of the challenges being identifying and removing phishing emails.

Organisations have procedures to prevent phishing emails from reaching their mail servers, and ultimately the system end-users, by using preventive measures such as automated scanning of incoming emails. Although this approach is important and used in every organisation, it does allow some phish through [9] because attackers usually adjust their tactics for the new defences. Therefore, as a best practice, organisations follow reactive measures to ensure that IT teams can contain the attack with minimal damage even if attackers succeeded to reach the system-end users. Responding to phishing attacks in a timely manner helps the IT reduce the number of victims and any potential organisational damage [10], [11], [12]. To be able to react to attacks successfully, IT teams need to learn about ongoing attacks through several sources such as security monitoring of abnormal behaviour, users' reporting, or external reports from other organisations [13], [14], [15]; though, the most common and valuable source is the users' reports [1] because phishers design their phishing emails to be devious with subtle differences between them to ensure a percent of the emails can bypass the filters; therefore, if all users reported phishing emails they receive, the IT staff would be able to catch all the variations in a campaign to develop a full picture of that campaign and understand its impact, sophistication and volume [1], [3], [15], [16], [17]. This action may require multiple teams in an organisation because phishing mitigation often requires configuration changes to a range of systems. Althobaiti et al. found that several teams deal with phishing reports including teams that manage user communication (Help Desk), Security, Accounts, Firewalls, Mail Relay, and Mail Storage [1].

In a typical phishing incident, users submit phishing reports to the IT general ticketing system. The reports are manually read through by the Help Desk or by a Security

team to ensure they are actually phishing [14], [18], [19]. One important procedure is to deal with phishing emails as campaigns because each campaign will have its own reactive measures that suits its impact and ramification factors [1]. In this procedural context, IT teams identify campaigns as the set of emails that were sent by the same attacker or a group of attackers who made an effort to create one sophisticated email but made many versions of that email while altering some aspects of sender id, email body, or email subject to bypass any blocks added and allow the campaign to last longer and harvest many victims [1], [16], [20]. Because of campaign variations, it is important that IT staff have access to a range of examples that covers all the variations. Therefore, the Help Desk staff collect sample emails of a phishing campaign that is meant to be representative of all the variations; however, the volume of reports the IT staff receive simultaneously about the same incident making it challenging to do so due to a lack of appropriate tools and the manual effort required in a timely manner [1] resulting in escalating unrepresentative sample which ends up in a partial remediation of the attack. Therefore, applying security remediation does not work always due to missing some of the variations. What the Help Desk staff need is an automatic approach to group the emails from the same campaign, so they can learn about the attack and easily find the common features in that campaign for a successful response.

B. EMAIL CLUSTERING

There has been substantial work on clustering emails. Applications include managing the email overloading problem by grouping emails into meaningful groups [21], [22], [23] such as subject-based folders [24] or personalised prioritisation [25].

In the context of phishing, clustering can form the basis of tools that support staff handling the problem mentioned in II-A by grouping reported phish into semi-campaigns so that staff can handle reports efficiently. We argue that, apart from classifying phishing emails into phishing versus benign decision [26], [27], [28], Help Desk staff would benefit from being presented with number of reported emails only. Once these reports have been manually labelled as phishing or benign, the remaining emails in that cluster can be automatically clustered into campaigns and flagged as phishing or benign. The method here aims to reduce the number of reports the Help Desk staff deal with and also help the other teams apply scripts to extract common features of the clusters.

In previous work, phishing emails have been clustered by profile [29], [30], [31], [32] where the approach attempts to group phishing emails written by a single individual or group [30], [33]. Work on phishing profiling was aimed to understand and observe attacker activities to better predict phishing emails [30], while some studies [34], [35] use it as a first step to improve the accuracy of a phishing/benign classifier. Seifollahi et al. [36] focused more on the authorship analysis and identifying the cybercriminal groups, while Zawoad et al. clustered emails in order to identify phishing

TABLE 1. Example of pair of emails that belong to the same campaign with different labels.

	Email 1	Email 2
Safety	phishing	unlabelled
From	Google <hans@xx.com.xx>	Google <hans@xx.com.xx>
Date	Fri 06 Sep 2019	Fri 06 Sep 2019
Subject	undeliverable messages	notification message
Body	g o o g l e support email...	g o o g l e service email...

attacks generated by off-the-shelf phishing kits [37]. Approaches have also looked at using semi-supervised phishing profiling to predict new spear phishing campaigns [38].

However, profiling is used mainly to identify phish authors, not campaigns. The distinction matters, because attackers can initiate several campaigns, each with different characteristics. We focus on clustering emails for identifying campaigns to help the IT staff with post-attack practices.

III. DATASET

A dataset of $n = 781,740$ emails was obtained from a major security services company located in the UK under a nondisclosure data sharing agreement. Emails in this dataset had been automatically labelled as phishing or left unlabelled based on comparisons with blacklists. Because campaigns often aim to confuse automated systems by using variations, there are definitely cases where some emails in a campaign are labelled as “phishing” while others are left as “unlabelled”. The emails in Table 1 show how two emails from the same campaign can get different safety labels.

To keep the number of emails small enough for partial human processing, we extracted two data slices (DS), each covering a time period of about six months. The first was from September 7th, 2019, to February 3rd, 2020. The second was from April 11th, 2019 to September 6th, 2019. Existing research shows that 50% of phishing campaigns last for 4 months and 25% last for 5 months [16], therefore, these time periods should be long enough to observe several variations of a phishing campaign.

We excluded emails that were missing mandatory headers, with the exception of the *TO* header as most of the emails had this information redacted for users’ privacy. We also excluded emails with parsing errors as they were likely corrupted during the redaction or transmission process. Table 2 shows the number of emails in both data slices, and the number of emails after the exclusions in every slice.

TABLE 2. Size of data slices (DS) used in experiments.

	dataset 1 (DS1)	dataset 2 (DS2)
Date	Sep 07 '19 - Feb 03 '20	Apr 11 '19 - Sep 07 '19
# of emails	67,729	68,568
Excluded	111	54
Included	67,618	68,514

IV. CLUSTERING

In this study we examine whether emails can be grouped into campaign-based clusters. Since campaigns change

dynamically over time, and we do not have access to a sufficiently large database of clusters with verified campaign labels, we use unsupervised clustering methods. We compared two clustering algorithms that have proven to be comparatively resilient to outliers (see Section IV-A) and tested them with seven different feature sets, defined in Section IV-B. For each algorithm, we also systematically varied a parameter that controls cluster size. Algorithms were tested on both data slices. For each algorithm, we systematically varied cluster size and feature sets. This results in 532 different solutions (i.e., 19 values of cluster size \times 7 feature sets \times 2 data slices \times 2 algorithms). The Scikit-learn library [39] was used for all experiments.

A. ALGORITHMS

We identified two candidate algorithms, Mean Shift [5] and DBSCAN [6], [40]. Both algorithms can automatically optimise the number of clusters generated. This is important, because the number of campaigns running at the same time varies. Both algorithms can handle numerical, categorical, and binary features as well as single features that can consist of a vector of values. Finally, both algorithms can deal with outliers, such as spear phishing attacks, where there may be only a single email. However, there is an important tradeoff between algorithms: While DBSCAN is faster, Mean Shift can identify clusters of varying size and may thus lead to more homogeneous clusters. Given that we use Mean Shift and DBSCAN without any changes, we only give a general outline below. We refer the readers to the original papers and the documentation of their implementation in Scikit-learn.

1) MEAN SHIFT

The Mean Shift [5] algorithm models each dataset as a combination of probability distribution. Each distribution is described by a kernel with a given bandwidth that is situated around a centre point. The number of kernels corresponds to the number of clusters, and data points are assigned to the nearest kernel. It is also possible to mark outliers as noise [41], [42]. The size of clusters in Mean Shift clustering is affected by a kernel bandwidth parameter which governs the spatial extent of the kernel's influence. A smaller bandwidth results in a tighter kernel that weighs nearby points more heavily, whereas a larger bandwidth leads to a broader kernel that encompasses a wider range of data points. Unfortunately, Mean Shift is computationally expensive ($O(N^2)$) to run on large data, which means that there is a trade-off between computational complexity and performance. To increase the algorithm speed, we set the bin seeding to true, which allocates discretised version of kernel points location with fewer seeds. We also allowed the algorithm to use all processors. For outliers, we did not cluster all the points to the nearest kernel so orphan emails (outliers) is labelled -1 .

In the context of clustering, Mean Shift works by updating candidates for centroids to be the mean of the points within

a given region. The position of the centroids is iteratively adjusted to eliminate near-duplicates and finalise the final centroids. Through successive iterations, data points shift towards regions of higher density within the feature space. This shifting process continues until convergence, at which point data points have gravitated towards the local maxima of the underlying distribution. For more details on Mean Shift procedure, please see [5], Section II.

2) DBSCAN

Unlike Mean Shift, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) [6], [40] follows several steps to generate clusters. The algorithm starts by determining core data points with a neighbourhood that includes at least n points that are less than ϵ apart from the core data point. These core data points serve as the seeds for cluster formation. If non-core data points are $< \epsilon$ away from a core data point, they are assigned to the cluster represented by this core data point by using euclidean distance function. The euclidean distance between a pair of row vector x and y is computed as:

$$\text{dist}(x, y) = \sqrt{\text{dot}(x, x) - 2 * \text{dot}(x, y) + \text{dot}(y, y)} \quad (1)$$

If they are further away from all core data points, they are labelled as noise. This ability to detect noise makes DBSCAN particularly suitable for identifying outliers [42]. It does not require a predefined set of clusters, so it can identify clusters that have different shapes, making it especially useful for datasets with complex structures. However, since the distance parameter ϵ is fixed, DBSCAN does not cope well with datasets where clusters can have varying similarity levels. For full pseudocode, please see [40], Algorithm 1. DBSCAN was chosen because it is computationally effective and has shown promising results for target recognition to find clusters of phishing web pages that mimic a legitimate webpage [28], [43]. Similar to Mean Shift, we used all the processors when running the algorithm.

The number of clusters generated by each algorithm depends on a single parameter that controls cluster sizes. For Mean Shift, this is the bandwidth parameter, while for DBSCAN, it is the distance ϵ parameter. After experimenting with random values on a smaller dataset, we examined performance for a total of 19 values, first values between [0.001, ..., 0.009] increasing in steps of 0.001, and then [0.01 ... 0.1], increasing in steps of 0.01.

B. FEATURES

Influenced by previous research on phishing detection and profiling [38], [44], [45], [46]—as well as our observation of several campaigns found in the dataset, six categories of mixed type features were used: time features ($n = 7$); subject features ($n = 3$); body features ($n = 16$); attachment features ($n = 4$); origin features ($n = 11$); recipient features ($n = 2$); and URL features ($n = 28$). These six groups of features were combined into seven feature sets summarised in Table 3.

1) TIME-BASED FEATURES

This feature category covers the time in which the phishing email was received. Phishing campaigns tend to be sent to organisation email addresses in batches within a short time frame [1], [16] making time-based features valuable for identification. Features are taken from the *DATE header* of the email. They are: date sent [38], time, day, month, year, weekday, and a derived binary feature (work day / non-work day). We added this feature since phishers might target working days as it is likely victims read the message before it is deleted [38].

2) SUBJECT-BASED FEATURES

This feature category is extracted from the email *SUBJECT header*. It covers number of characters [38], number of white spaces, and the vector of Term Frequency - Inverse Document Frequency (TF-IDF) values of all words in the subject.

3) BODY-BASED FEATURES

This feature category was derived from the plain text part and the HTML part of the email object. In order to check the web technology used, we computed the types and numbers of email elements, presence and number of images, presence and number of URLs, and presence of HTML tags, scripts, and CSS specifications [45], [46], [47]. We then removed all HTML tags and other scripts as well as links to obtain the pure body text. The text was converted into a bag of words. We used Latent Semantic Analysis to extract the top ten terms describing the email's content [38], [45]. We also computed the number of lines, number of words, and average word length [38]. While prior research focused on whether an email contain a greeting line or not [47], from our observations we found that several campaigns follow the same greeting type. Therefore, we added a feature describing the greeting type (style of greeting, such as hi, hello, and dear; checking whether greeting is followed by recipient name, username or email address).

4) ATTACHMENT-BASED FEATURES

This feature category concerns the email attachments. We determined whether the email has an attachment, how many attachments the email has [38], [46], and attachment size and type [38]. This information indicates if the attacker distributes the same files within a campaign.

5) ORIGIN-BASED FEATURES

The origin feature category is mostly about the sender of the email. This can be either the attacker themselves or the compromised accounts. We extracted name and email address from both the *FROM header* and the *RECEIVED header*. We also checked whether the email from the *RECEIVED header* matches the one in the *FROM header* in order to detect spoofed FROM addresses. This information can indicate the impersonated identity and details on the origin of phishing campaign. We extracted the sender IP [38] and

relevant domain information such as the domain from both headers [38], domain registrar, domain registration date and the registrar location [38]. This provides information about the attacker origin and whether they used a public service or compromised accounts to send the email.

6) RECIPIENT-BASED FEATURES

Recipient features concern the target users which only includes recipient names and recipient counts. Other information that has been shown to be effective at identifying the target characteristics [38] was excluded as most of the information we have about recipients was redacted for anonymity reasons.

7) URL-BASED FEATURES

URL-based features are one of the most important features in phishing detection [48], [49], [50]. In this work we excluded any feature that requires visiting the link, because it takes a long time, and for older emails, the links probably were taken down or changed. Features in the URL category include the domain names, hostnames, domain categories, location of domain registrar, subdomain count, and hyphen count. We also computed binary features that reflect whether at least one URL in the email has an Extended Validation Certificate (EV) that validates the owner of the domain, an extra http and Top-Level Domain (TLD), a web-host domain, a "@" symbol, non-ASCII characters, whether it has typos comparing to top 10,000 popular domains, whether it is similar to top targeted domains on PhishTank and whether one of the subdomains contains a popular domain on PhishTank [51].

For emails with several URLs, we counted the number of URLs with an IP address, the number of different domains, number of short URLs, and number of blacklisted links. In the case of hyperlinks, we checked whether the visual link presented in the email directed to the same URL [46], [47] and checked whether there was a link under a text such as *click here*. For the domain information, we collect the registration dates of the oldest and the most recent domains, the minimum PageRank and popularity, and the maximum PageRank and Popularity for the list of URLs.

8) FEATURE SETS

We then combined the six feature categories into seven feature sets (FS) as shown in Table 3 to examine which set better contributes to the algorithm performance [47]. FS1 includes all the features while FS2 and FS7 consist of features that appeared to be particularly relevant based on examination of a small sample of the dataset. FS3 focuses on URL based features because they are one the most common indicators of phishing emails [52] and most used features in phishing detection research [53], [54]. To ensure we can catch as much of the campaign variations as possible, we also include the origin features which also would help capture campaigns without URLs. Text features showed promising results in phishing detection [45]; thus we have two categories of the

TABLE 3. Summary of features in every feature set (FS).

FS	Features
FS 1	All the features of FS2-FS7
FS 2	A subset of set 1: Time, Subject, Attachment, Body and URL features except hostnames, domain location and typos. Origin features except the sender email addresses and number of recipients from Recipient features
FS 3	URL features and whether the email has URLs and number of them Origin features (except domain age)
FS 4	Email text, email subject and topic features
FS 5	Email subject, message text, element types, topic features and attachment type, Recipient names and origin features except IP, domain age and headers match, URLs hostnames, URLs domains and domain's registered locations
FS 6	Subject length and count of spaces and attachment count and size Date, Body, origin, attachments and URL features except set 4
FS 7	Subject features, topic features and Body features except the message text, Sender name, location and domain category and URL features.

textual features FS4 and FS5. FS5 has all the textual features, whereas FS4 has fewer number of features to explore its effectiveness without consuming memory and time. For FS6 we considered numeric and categorical features in case the algorithms work better with smaller dimensionality.

Due to the size of the features, we reduced dimensionality of the feature vectors using Singular Value Decomposition.

C. EVALUATION

Cluster solutions were evaluated using two metrics, silhouette and homogeneity.

1) SILHOUETTE COEFFICIENT SCORE

The silhouette index is an internal validation metric for measuring how the clusters are formed with respect to their compactness and separation. Silhouette metric determines for each data point whether it is more similar to the cluster C_j it has been assigned to than to the clusters $C_{i \neq j}$.

For each data point $i \in C_i$, $a(i)$ is defined as:

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, i \neq j} d(i, j) \quad (2)$$

For each data point $i \in C_I$, $b(i)$ is defined as:

$$b(i) = \min_{J \neq I} \frac{1}{|C_J|} \sum_{j \in C_J} d(i, j) \quad (3)$$

Silhouette value of one data point i is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \quad \text{if } C_I > 1 \quad (4)$$

The index value varies between $[-1, 1]$. A value close to 1 indicates a good match while a value close to -1 indicates that data points are badly matched to clusters. Silhouette is a standard metric for evaluating the quality of a clustering solution.

2) HOMOGENEITY SCORE

The homogeneity score is an external validation metric to measure how the cluster labels match externally provided

labels [55]. A clustering solution is homogeneous if all of its clusters contain only samples belonging to a single class. Therefore, more homogeneous cluster solutions are more likely to be useful to human users. Ideally, clusters should either contain only benign emails or emails that are part of the same phishing campaign. We did not penalise solutions where campaigns were spread over several clusters. The score ranges between $[0.0, 1.0]$, where 1.0 stands for perfectly homogeneous labelling.

For homogeneity score, we need labelled data. It was not possible to assign each of the 10K+ putative phishing emails to campaigns; therefore, we manually identified 10 campaigns which were used to establish homogeneity. We first identified five random candidate emails per data slice. For each of these candidate emails, we found at least four further emails with similar characteristics. Then, we searched the dataset to find more emails that shares the sender names, sender emails, email subjects or topic features for each of the $2 \times 5 = 10$ campaigns and manually determined whether these emails were part of the campaign or not. The campaigns identified are summarised in Table 4. Each campaign has variations in one or many features, and 6 of the 10 campaigns stretched across both data slices.

TABLE 4. Summary of the manually labelled campaigns. The first column indicates the campaign impersonated authority or its main topic, whereas the second column indicates the number of emails per data slice and the third indicates the common features in each campaign.

Campaign	No. of Emails		Characteristics
	Slice1	Slice2	
Royal Bank	2	91	Spoofed addresses
Driving License	118	–	Identical sender name
Vehicle Tax Refund	–	6	Only variations in sender email
Revenue Agency	61	204	Identical sender name and phrases
Dream jobs	73	48	Unique common keywords
Free Package	93	–	Identical subject, Bitcoin
Accountancy Services	62	109	Identical sender name and phrases
Document order	269	1	FYI document
Recovery Email	15	64	Different subject and sender
Facebook Payment	161	–	Slight variations in email body

V. EXPERIMENTAL EVALUATION

Using Silhouette and Homogeneity scores, we detail the process of finding the most effective algorithm and parameters to better group emails into campaigns in order to fulfil the purpose of this study.

A. CHOICE OF ALGORITHM

We applied DBSCAN and Mean Shift to all seven feature sets and both data slices. The resulting silhouette scores are found in Fig. 1, while the homogeneity scores against the labelled subset are shown in Fig. 2.

We clearly see that Mean Shift performs consistently well, while some combinations of feature sets lead to very poor performance with DBSCAN. On the silhouette score, Mean Shift outperformed DBSCAN with values ranging between 0.30 and 0.67 (*Mean* = 0.40, *Median* = 0.36), whereas DBSCAN's values range between -0.63 and 0.42 (*Mean* = -0.16 , *Median* = -0.35). Looking at the homogeneity

score, we find that for Mean Shift, the distribution of homogeneity was (*Min* = 0.67, *Max* = 0.94, *Mean* = 0.81, *Median* = 0.80), whereas for DBSCAN, we saw much wider variation (*Min* = 0.31, *Max* = 0.93, *Mean* = 0.63, *Median* = 0.52). Overall, across all combinations of feature and datasets, Mean Shift consistently produces better results than DBSCAN. This might be due to the variation in campaign similarity demonstrated in Table 4. Since DBSCAN uses a fixed threshold ϵ to determine cluster boundaries, it is less well equipped to deal with such variation. Therefore, in the remainder of this paper, we will only report Mean Shift results.

B. PERFORMANCE OF FEATURE SETS

We summarise the evaluation of the cluster models for each data slice in Table 5. Homogeneity scores are high, which indicates that each cluster is likely to only contain data points from a single phishing campaign. The silhouette score, on the other hand, assesses the separation of clusters. However, most silhouette scores are substantially lower. This suggests that most feature sets result in clusters that are not well separated in the space defined by the feature set. FS3 has the best Silhouette score 0.65 and 0.67 for DS1 and DS2 respectively, followed by FS7. FS7 and FS3 both have the highest homogeneity score 0.94 for DS1, followed by FS4 (0.86), and FS7 has the highest score for DS2 (0.88), followed by FS3 (0.87).

Overall, FS3 performs well on silhouette and homogeneity scores. FS7 also performs acceptably on both measures. Of the three feature sets, FS3 is the smallest and is also closest to the features used in phishing detection. In contrast, FS7 is larger, more complex, and more expensive to compute, as it requires Latent Semantic Analysis of the email body and/or TF-IDF vectorization of the email subject. Therefore, in the remainder of this paper, we will use the optimal Mean Shift solution for feature set F3.

TABLE 5. Silhouette and Homogeneity Scores for Mean Shift algorithm for every feature set. Best value for each data slice in bold and second best value in italic.

FS	Silhouette		Homogeneity	
	DS1	DS2	DS1	DS2
FS1	0.31	0.3	0.74	0.74
FS2	0.3	0.3	0.74	0.8
FS3	0.65	0.67	0.94	0.87
FS4	0.42	0.38	0.86	0.83
FS5	0.38	0.33	0.79	0.67
FS6	0.34	0.34	0.78	0.81
FS7	0.47	0.47	0.94	0.88

C. ANALYSIS OF BEST SOLUTION

We explored the clustering solution for FS3 for both data slices to get a sense of how potentially helpful they might be to IT staff.

We identified 2720 clusters with varying size (*Mean* = 22.83, *Median* = 8, *Max* = 1108) for DS1 and 3380 clusters for DS2 (*Mean* = 18.71, *Median* = 6, *Max* = 943). In other words, this approach can cluster \approx 60K emails into \approx 6K

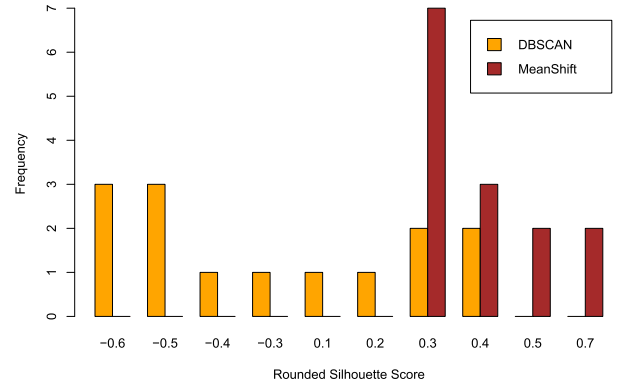


FIGURE 1. The distribution of Silhouette scores based on the clustering algorithm used.

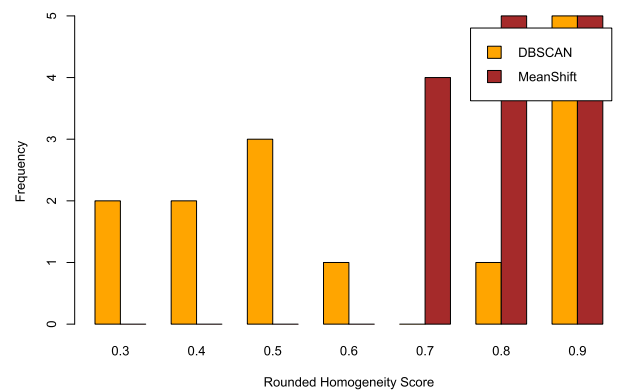


FIGURE 2. The distribution of Homogeneity scores based on the clustering algorithm used.

clusters. If the clusters are embedded in a system that requires IT staff to only need to check one or two representative emails per cluster, the time needed to identify phishing emails and associated campaigns can be reduced by 80–90%. The range in cluster size also suggests that even a few well chosen clusters, such as the 10-20 largest, could greatly save staff time, since some clusters appear to be long-running and large.

The silhouette and homogeneity scores tell us that the resulting clusters are well formed. Interestingly FS3 does not include subject line which we know is often used by IT teams to quickly judge if phishing reports are from the same campaign or not along with other user-visible data like from addresses which do appear in FS3 [1], [18]. To better understand how these clusters might look to IT staff we analysed subject line and FROM address variations in the clusters.

We find that subject lines do indeed often vary within clusters. Only $DS1 = 322$ (11.84%) and $DS2 = 491$ (14.53%) clusters contained a single subject line meaning \approx 70% of clusters contain subject line variations. To understand the amount of subject line variation we computed the number of duplicated subject lines per cluster (count of unique subject lines subtracted from total number of emails) and then normalised by dividing the total number of emails in that cluster.

This results in a number between 0 and 1 where 0 means that no two emails in a cluster had the same subject and 1 means that the whole cluster had the same subject. We find a mean of 0.47 (DS1, $\sigma = 0.41$) and 0.49 (DS2, $\sigma = 0.42$) suggesting that there is indeed a wide variation between clusters as shown by the standard deviations of nearly a half.

The *FROM* address is also used by IT staff to quickly determine if phishing reports are likely from the same campaign. Since FS3 includes email origin features like *FROM*, it is somewhat surprising that only $\approx 20\%$ of clusters (DS1, $n=582$, 21.40% and DS2, $n=805$, 23.82%) contain only one sender name and email. We computed the number of duplicated *FROM* addresses in a cluster divided by the total number of emails in that cluster. We found that a mean of 0.67 (DS1, $\sigma = 0.38$) and 0.68 (DS2, $\sigma = 0.38$) suggesting fewer unique *FROM* addresses within clusters compared to subject lines. This makes some sense given that FS3 uses *FROM* address as a clustering feature.

VI. EXPERT EVALUATION

Silhouette and Homogeneity scores are excellent metrics to validate the consistency within the clusters but we need another method to evaluate the clusters against whether IT teams can consider each cluster a phishing campaign or a subset of a phishing campaign. We used a questionnaire structure where each participant sees a pair of emails from the same cluster and decides whether they belong to the same campaign or not.

A. STUDY DESIGN

We used an online questionnaire structure that started with the study description and a definition of a phishing campaign. We explained that a campaign is a series of phishing attacks that are likely performed by a phishing group who is impersonating a specific authority or random people while following a similar set of tactics within a short period of time [38], [56], [57]. We also highlighted that the emails in the survey may or may not be phishing emails as our dataset includes some real emails that have been reported such as advertising emails and newsletters. To make sure they perceived the concept accurately, we showed them three examples of pair of emails that belong to the same campaign and explained why they belong to the same campaign. The first pair of emails has the same sender name, email address, and subject but were delivered on two different days in the same week. In the second example, we showed them two emails with the same email address but different sender names, subjects and bodies with common phrases such as “important spam report”. In the third example, they were shown two dissimilar emails that clearly belong to the same campaign as they both spoofed the same authority, had tax-related topics and contained similar keywords.

We showed participants 120 pairs of emails selected as described below. For each pair, we asked them whether the two emails belong to the same campaign or not. Each email contained the: *sender name*, *from address*, *email delivery*

date, *email subject*, and the first 300 characters of the *email body*. Pair of emails were shown to participants in a random order.

We selected the pairs for human labelling from the clustering solution generated by the Mean Shift cluster with Feature set FS3. We then randomly selected email pairs using a 2×3 sampling design based on if they were from the same cluster or not and if they were very similar, moderately similar, or dissimilar. Consequently 60 email pairs were selected where both emails were from the same cluster, and 60 where they were from different clusters. Within those 60, we sampled such that 20 were from each of the three levels of similarity. To identify the similarity levels, we used the three variables: sender, email subject, and date. These were chosen because IT teams are known to use them when reviewing reports [1]. Very similar emails differed in only one variable; moderately similar emails differed in two variables; and dissimilar emails had all three variables different. Email pairs were randomly selected from all possible emails such that these constraints were matched.

B. SURVEY RESULTS

Due to the NDA agreement, we recruited 5 participants from our lab for a one-hour study on phishing campaigns. All of the participants have previously seen talks about our study that explained what phishing campaigns are and three of them are phishing experts who work on similar projects. The type of participants selected in this study reflects the type of staff in the IT Help Desk. They have enough knowledge of phishing and aware of the organisational phishing handling procedures. In practice, any senior Help Desk staff can identify any set of groups as a campaign [1]; which is represented in this study by recruiting 5 annotators with a reasonable agreement between them.

All the participants were asked to classify all of the 120 pair of emails as being from the same campaign or not. We computed the agreement between the participants using Fleiss' Kappa and found that they strongly agreed with a kappa value of 0.87.

We computed the final labels based on what the majority of annotators agreed on as seen in Table 6. For within cluster emails, we found that participants always agree with the identified clusters when the emails are very similar and moderately similar for the 20 emails in each group. For dissimilar emails, participants disagreed with the clustering for about half of the emails (11/20). Our findings suggest that very similar and moderately similar emails from within the cluster are likely to be from the same campaign.

For the 60 between cluster emails, emails were picked from two different clusters. Participants labelled all the very similar emails as being from the same campaign, i.e. even if emails are not grouped in the same cluster, they can be from the same campaign. For moderately similar emails, they labelled few as being from the same campaign (3/20) and all the dissimilar emails were correctly labelled as not from the same campaign (0/20), i.e. similar emails are likely to be from the

TABLE 6. The confusion matrix to summarise the clustering performance based on participants decisions. Each table represents a level of similarity.

Almost Similar		
	Actual campaign	Actual not campaign
Predicted campaign	20	0
Predicted Not campaign	20	0
Moderately Similar		
	Actual campaign	Actual not campaign
Predicted campaign	20	0
Predicted Not campaign	3	17
Not Similar		
	Actual campaign	Actual not campaign
Predicted campaign	9	11
Predicted Not campaign	0	20

same campaign, whereas moderately similar and dissimilar emails from between clusters are unlikely to be from the same campaign.

The true positive and true negative values were computed. We found that the true positive is 49 out of 60. i.e. total number of agreement with the algorithm decision of assigning the two emails to the same cluster. The false positive of assigning two emails from different campaigns to the same cluster was 11 out of 60. We then computed the precision (0.82), recall (0.57) and F-score (0.68) indicating that the algorithm correctly predicted most of the within cluster emails and more than the half of the between cluster emails were correctly predicted.

VII. DISCUSSION

This paper investigated the feasibility of grouping reported phishing emails into clusters as a first step to designing a tool that helps IT staff identify campaigns. To the best of our knowledge, this is the first study that leverages unsupervised clustering to help identify campaigns. This study is therefore complementary to the clustering work discussed in Section II-B, which focuses more on identifying attackers.

We find that Mean Shift with a feature set consisting of URL-based features and origin-based features (FS3) and a small kernel bandwidth is the most effective setting for identifying potential campaigns. It produced well-separated and homogeneous clusters. One possible reason is that email and URL domains are expensive to obtain at scale making phishers attempt to manipulate them instead of registering new ones [58].

The small bandwidth resulted in a larger number of clusters compared to the findings from other studies [30] because, in line with our goal to identify campaigns, we prioritised homogeneity. Larger clusters make sense when the goal is to identify and profile prolific phishers but in our case we identify campaigns that may be originated by the same group.

A. FEASIBILITY OF CLUSTERING FOR CAMPAIGN IDENTIFICATION

Currently, Help Desk staff handle every phishing report individually and once they identify a phishing email, they remove

all the emails with the same sender, subject and date which does not remove all the emails in that campaign. In our work, we found that our clusters contain variations in the either the subject, sender and delivery date.

In a separate user study with expert annotators (Section VI), we studied whether our result can be feasible in an organisational environment or not. The goal of this study is to examine whether the clustered emails are seen to be from the same campaign or not.

Although the study analysed the true and false positive and negative in the clustering result, our proposed solution only focuses on the true positive and negative as the goal is to reduce the work load on the IT staff. It is well tolerated for our solution to split a campaign into two or many clusters but not to add two campaigns in one cluster. Therefore, for the true positive and negative, our result found that similar and moderately-similar emails (judging by date, subject, and sender) from the same cluster are most likely to belong to the same campaign. This result shows that the clusters identified by our best performing solution fit well with criteria that IT staff are using.

B. IMPLICATIONS FOR DESIGN

Clustering incoming phishing reports has the potential to greatly help the many different IT-related teams across an organisation coordinate their response to identified attacks. Unlike issues such as firewall misconfigurations, phishing touches on many aspects of an organisation and is an example of a distributed cognition process [1] where many people work together to solve a problem. To give an example, users might first report to a Help Desk who review reports and then escalate them to various email server teams to remove the email from inboxes as well as stop it from coming in across the mail servers. Firewall teams may identify any malicious URLs or IP addresses in the email and update firewall rules to block them. Account management teams may also need to scan for potential compromise and reset passwords. Each team needs slightly different information about the campaign, and each team has a different set of responsibilities. A current common practice to help handle the scale of reports is for the help desk to identify a couple of useful looking reports early on and escalate them for the other teams to use [1] which facilitates fast action and response. But it prevents the various teams from being able to understand the full variation range of the campaign they are addressing, including variations which may be added later.

Automating the process can be very helpful to the distributed teams working on a phishing campaign case. The proposed solution can be useful in a tool that automatically added new reports to an escalated ticket as they came in and scanned them for potential new variations, such as sending from a new mail server or using a new URL domain.

Because clustering is rarely completely accurate, we anticipate that IT staff will have to handle some cases where emails are not clustered correctly. However, based on our user study, we can see that these errors are not random and are much

more likely to occur in cases where several key variables differ between emails. Automatically removing such emails from a cluster is a poor idea, because they are also potentially the emails exhibiting the widest variations and are therefore they are most helpful in making sure that the whole cluster is mitigated against. Using this observation, it may be possible to design an interface for staff that highlights the emails that are most likely to be incorrectly categorised in a way that allows them to judge if the email is miscategorised or if it is an example of an unexpected variation that needs more attention. Such a feature would both greatly help IT staff as well as potentially provide training data back to a clustering system. It is therefore worth potentially exploring in future work.

VIII. LIMITATIONS AND FUTURE WORK

There are several limitations to our study, which need to be addressed in future work.

First, the dataset was collected from a specific organisation over two somewhat short time frames. While the data is designed to be fairly representative of what current phishing looks like, we note that it was collected using a sinkhole type approach by collecting data from email addresses that are no longer used. The organisation that gave us the data ran automated scripts to filter out spam and non-phishing but these filters were not 100% accurate. Depending on the quality of the spam filter, this approach might also have caught some phishing attempts. Thus, our approach needs to be replicated within an organisation on actual data during several different time frames. Though we do note that the mix of phishing and non-phishing is also representative of our goal use case where users are unlikely to be 100% accurate in their reporting of phishing.

Second, in order to assess its effectiveness in practice, clustering needs to be integrated meaningfully into the workflow of IT teams. For example, our proposed approach can be integrated into the current reporting systems to automatically generate one report of verified clusters instead of showing all the user reports individually. The report then can summarise variations based on what each IT team needs. For example, it can show a list of sender IPs to the team in charge of the mail server so they can update the spam filters accordingly, or a list of email addresses, subjects and dates for the team managing the Exchange server to delete the emails based on. The list of variations will also help Incident Response teams to understand the range of a campaign's variations and react to it according to the deceptiveness level of the campaign [16] and the scale of variation [1].

In this context, the choice of clustering algorithm used should be revisited to see whether less computationally expensive approaches can be used instead of Mean Shift. Acceptable processing speed needs to be investigated in the context of deployment in an actual solution, with a better understanding of which reaction times to campaigns are considered timely [1].

Integration of our clustering approach with a classifier that distinguishes between phishing and benign emails is crucial for successful deployment. Since we focus on small, homogeneous clusters, we suggest that all emails in a cluster are likely to share the same label. Unlike content based features, the features in our top-performing dataset, FS3, are strong and hard to mimic by the attacker such as the sender IP, URL domain, and domain registrar. Second, similar features have been used in several research projects to recognise phishing emails [59] with accuracy of up to 96%.

Finally, due to the size of the dataset, it is not possible to annotate all emails with ground truth phishing / benign and campaign labels. In addition, Our data does not have verified benign emails but only verified blacklisted emails; thus, we cannot measure how accurate our approach would be with the safety labels. We suggest that future labelling efforts should focus on edge cases. In a larger user study, where unsupervised clustering is combined with a phishing/benign classifier, the emails at the centre of large clusters can be investigated to see whether they might be good indicators of potential campaigns.

IX. CONCLUSION

We explored the feasibility of using clustering to identify phishing campaigns using reported phishing emails from a large security company. We applied two algorithms with different sets of features on two datasets. We found that Mean Shift algorithm outperformed DBSCAN for clustering phishing campaigns with the feature set that composes of the email sender, subject, body, and URL based features. Using two validation metrics and experts evaluation, the results indicate the potential for assisting IT teams in handling the complexity and large scale of phishing reporting associated with attacks.

ACKNOWLEDGMENT

The researchers would like to acknowledge Deanship of Scientific Research, Taif University for Funding this work, University of Edinburgh for providing the resources needed for running the study, and the security company that offered the dataset for the study. The authors would like to thank the TULiPS Laboratory at the University of Edinburgh, for helpful discussions and feedback.

REFERENCES

- [1] K. Althobaiti, A. D. G. Jenkins, and K. Vaniea, "A case study of phishing incident response in an educational organization," *Proc. ACM Hum. Comput. Interact.*, vol. 5, no. CSCW2, p. 338, 2021, doi: [10.1145/3476079](https://doi.org/10.1145/3476079).
- [2] H. Bo, W. Wei, W. Liming, G. Guanggang, X. Yali, L. Xiaodong, and M. Wei, "A hybrid system to find & fight phishing attacks actively," in *Proc. IEEE/WIC/ACM Int. Conferences Web Intell. Intell. Agent Technol.* Lyon, France: IEEE Computer Society, vol. 1, Aug. 2011, pp. 506–509, doi: [10.1109/WI-IAT.2011.94](https://doi.org/10.1109/WI-IAT.2011.94).
- [3] A. Shah, R. Ganesan, S. Jajodia, and H. Cam, "Understanding tradeoffs between throughput, quality, and cost of alert analysis in a CSOC," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 5, pp. 1155–1170, May 2019, doi: [10.1109/TIFS.2018.2871744](https://doi.org/10.1109/TIFS.2018.2871744).

- [4] A. J. Burns, M. E. Johnson, and D. D. Caputo, "Spear phishing in a barrel: Insights from a targeted phishing campaign," *J. Organizational Comput. Electron. Commerce*, vol. 29, no. 1, pp. 24–39, Jan. 2019, doi: [10.1080/10919392.2019.1552745](https://doi.org/10.1080/10919392.2019.1552745).
- [5] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 5, pp. 603–619, May 2002, doi: [10.1109/34.1000236](https://doi.org/10.1109/34.1000236).
- [6] M. Ester, H. Kriegl, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD)*. Palo Alto, CA, USA: AAAI Press, 1996, pp. 226–231. [Online]. Available: <https://dl.acm.org/doi/10.5555/3001460.3001507>
- [7] Department for Digital Culture Media & Sport, "Official statistics cyber security breaches survey 2020—Chapter 5: Incidence and impact of breaches or attacks," Nat. Cyber Security Centre, Colorado Springs, CO, USA, Tech. Rep. 5, May 2020, Accessed Jan. 2021. [Online]. Available: <https://www.gov.uk/government/publications/cyber-security-breaches-survey-2020/cyber-security-breaches-survey-2020>
- [8] *2020 Data Breach Investigations Report*, Verizon Trademark Services LLC, New York, NY, USA, Jun. 2020. [Online]. Available: <https://vz.to/3vKNI1K>
- [9] A. Muneer, R. F. Ali, A. A. Al-Sharai, and S. M. Fati, "A survey on phishing emails detection techniques," in *Proc. Int. Conf. Innov. Comput. (ICIC)*, Nov. 2021, pp. 1–6, doi: [10.1109/ICIC53490.2021.9692960](https://doi.org/10.1109/ICIC53490.2021.9692960).
- [10] *2019 Dataenterprise Phishing Resiliency and Defense Report Breach Investigations Report*, Verizon Trademark Services LLC, New York, NY, USA, 2019, Accessed: Jun. 2020. [Online]. Available: <https://vz.to/2RukvJC>
- [11] R. Werlinger, K. Muldner, K. Hawkey, and K. Beznosov, "Preparation, detection, and analysis: The diagnostic work of IT security incident response," *Inf. Manage. Comput. Secur.*, vol. 18, no. 1, pp. 26–42, Mar. 2010, doi: [10.1108/09685221011035241](https://doi.org/10.1108/09685221011035241).
- [12] F. B. Kokulu, A. Soneji, T. Bao, Y. Shoshitaishvili, Z. Zhao, A. Doupe, and G.-J. Ahn, "Matched and mismatched SOCs: A qualitative study on security operations center issues," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, London, U.K., Nov. 2019, pp. 1955–1970, doi: [10.1145/3319535.3354239](https://doi.org/10.1145/3319535.3354239).
- [13] G. Grispos, W. B. Glisson, D. Bourrie, T. Storer, and S. Miller, "Security incident recognition and reporting (SIRR): An industrial perspective," in *Proc. 23rd Americas Conf. Inf. Syst. (AMCIS)*. Boston, MA, USA: Association for Information Systems, Aug. 2017, pp. 1–10. [Online]. Available: <http://aisel.aisnet.org/amcis2017/InformationSystems/Presentations/15>
- [14] E. Koivunen, "Why wasn't I notified: Information security incident reporting demystified," in *Information Security Technology for Applications (Lecture Notes in Computer Science)*, vol. 7127. Espoo, Finland: Springer, Oct. 2010, pp. 55–70, doi: [10.1007/978-3-642-27937-9_5](https://doi.org/10.1007/978-3-642-27937-9_5).
- [15] S. Metzger, W. Hommel, and H. Reiser, "Integrated security incident management—concepts and real-world experiences," in *Proc. 6th Int. Conf. IT Secur. Incident Manage. IT Forensics*, Stuttgart, Germany: IEEE Computer Society, May 2011, pp. 107–121, doi: [10.1109/IMF.2011.15](https://doi.org/10.1109/IMF.2011.15).
- [16] A. van der Heijden and L. Allodi, "Cognitive triaging of phishing attacks," in *Proc. 28th USENIX Security Symp.* Santa Clara, CA, USA: USENIX Association, 2019, pp. 1309–1326. [Online]. Available: <https://www.usenix.org/conference/usenixsecurity19/presentation/van-der-heijden>
- [17] A. Ahmad, J. Hadgkiss, and A. B. Ruighaver, "Incident response teams—challenges in supporting the organisational security function," *Comput. Secur.*, vol. 31, no. 5, pp. 643–652, Jul. 2012, doi: [10.1016/j.cose.2012.04.001](https://doi.org/10.1016/j.cose.2012.04.001).
- [18] M. L. Jensen, A. Durcikova, and R. T. Wright, "Combating phishing attacks: A knowledge management approach," in *Proc. 50th Hawaii Int. Conf. Syst. Sci. (HICSS)*. Hilton Waikoloa Village, HI, USA: AIS Electronic Library (AISEL), Jan. 2017, pp. 1–10. [Online]. Available: <http://hdl.handle.net/10125/41681>
- [19] M. Husák and J. Cegan, "PhiGARo: Automatic phishing detection and incident response framework," in *Proc. 9th Int. Conf. Availability, Rel. Secur. Fribourg*, Switzerland: IEEE Computer Society, Sep. 2014, pp. 295–302, doi: [10.1109/ARES.2014.46](https://doi.org/10.1109/ARES.2014.46).
- [20] E. Lastdrager, P. Hartel, and M. Junger, "Poster: Apaté: Anti-phishing analysing and triaging environment," in *Proc. 36th IEEE Symp. Security and Privacy*. United States: IEEE Computer Society, May 2015, pp. 1–2. [Online]. Available: https://www.ieee-security.org/TC/SP2015/posters/paper_58.pdf
- [21] Y. Xiang, "Managing email overload with an automatic nonparametric clustering system," *J. Supercomput.*, vol. 48, no. 3, pp. 227–242, Jun. 2009, doi: [10.1007/s11227-008-0216-y](https://doi.org/10.1007/s11227-008-0216-y).
- [22] A. Sharaff and N. K. Nagwani, "ML-EC2: An algorithm for multi-label email classification using clustering," *Int. J. Web-Based Learn. Teaching Technol.*, vol. 15, no. 2, pp. 19–33, Apr. 2020, doi: [10.4018/IJWLTT.2020040102](https://doi.org/10.4018/IJWLTT.2020040102).
- [23] M. G. Armentano and A. A. Amandi, "Enhancing the experience of users regarding the email classification task using labels," *Knowl.-Based Syst.*, vol. 71, pp. 227–237, Nov. 2014, doi: [10.1016/j.knsys.2014.08.007](https://doi.org/10.1016/j.knsys.2014.08.007).
- [24] I. Alsmadi and I. Alhami, "Clustering and classification of email contents," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 27, no. 1, pp. 46–57, Jan. 2015, doi: [10.1016/j.jksuci.2014.03.014](https://doi.org/10.1016/j.jksuci.2014.03.014).
- [25] S. Yoo, Y. Yang, F. Lin, and I.-C. Moon, "Mining social networks for personalized email prioritization," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Paris, France, Jun. 2009, pp. 967–976, doi: [10.1145/1557019.1557124](https://doi.org/10.1145/1557019.1557124).
- [26] K. Georgala, A. Kosmopoulos, and G. Paliouras, "Spam filtering: An active learning approach using incremental clustering," in *Proc. 4th Int. Conf. Web Intell., Mining Semantics (WIMS14)*, Thessaloniki, Greece, Jun. 2014, p. 23, doi: [10.1145/2611040.2611059](https://doi.org/10.1145/2611040.2611059).
- [27] D. DeBarr, V. Ramanathan, and H. Wechsler, "Phishing detection using traffic behavior, spectral clustering, and random forests," in *Proc. IEEE Int. Conf. Intell. Secur. Informat.*, Seattle, WA, USA, Jun. 2013, pp. 67–72, doi: [10.1109/ISI.2013.6578788](https://doi.org/10.1109/ISI.2013.6578788).
- [28] G. Liu, B. Qiu, and L. Wenyin, "Automatic detection of phishing target from phishing webpage," in *Proc. 20th Int. Conf. Pattern Recognit. Istanbul, Turkey: IEEE Computer Society*, Aug. 2010, pp. 4153–4156, doi: [10.1109/ICPR.2010.1010](https://doi.org/10.1109/ICPR.2010.1010).
- [29] R. Hidayat, I. T. R. Yanto, A. A. Ramli, and M. F. M. Fudzee, "Similarity measure fuzzy soft set for phishing detection," *Int. J. Adv. Intell. Inform.*, vol. 7, no. 1, pp. 101–111, 2021. [Online]. Available: <http://mail.ijain.org/index.php/IJAIN/article/view/605>
- [30] I. R. A. Hamid and J. H. Abawajy, "Profiling phishing email based on clustering approach," in *Proc. 12th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun.*, Jul. 2013, pp. 628–635.
- [31] J. Yearwood, M. Mammadov, and D. Webb, "Profiling phishing activity based on hyperlinks extracted from phishing emails," *Social Netw. Anal. Mining*, vol. 2, no. 1, pp. 5–16, Mar. 2012, doi: [10.1007/s13278-011-0031-y](https://doi.org/10.1007/s13278-011-0031-y).
- [32] I. R. A. Hamid, N. A. Samsudin, A. Mustapha, and N. Arbaiy, "Dynamic traceback strategy for email-borne phishing using maximum dependency algorithm (MDA)," in *Proc. 2nd Int. Conf. on Soft Computing and Data Mining (SCDM)*, in *Advances in Intelligent Systems and Computing*, vol. 549. Bandung, Indonesia: Springer, 2016, pp. 263–273, doi: [10.1007/978-3-319-51281-5_27](https://doi.org/10.1007/978-3-319-51281-5_27).
- [33] L. Ma, J. Yearwood, and P. Watters, "Establishing phishing provenance using orthographic features," in *Proc. eCrime Researchers Summit*, Tacoma, WA, USA, Sep. 2009, pp. 1–10, doi: [10.1109/ECRIME.2009.5342604](https://doi.org/10.1109/ECRIME.2009.5342604).
- [34] I. R. A. Hamid and J. H. Abawajy, "An approach for profiling phishing activities," *Comput. Secur.*, vol. 45, pp. 27–41, Sep. 2014, doi: [10.1016/j.cose.2014.04.002](https://doi.org/10.1016/j.cose.2014.04.002).
- [35] A. Hamza and H. Moetque, "Feature weight optimization mechanism for email spam detection based on two-step clustering algorithm and logistic regression method," *Int. J. Adv. Comput. Sci. Appl.*, vol. 8, no. 10, pp. 420–429, 2017, doi: [10.14569/IJACSA.2017.081054](https://doi.org/10.14569/IJACSA.2017.081054).
- [36] S. Seifollahi, A. Bagirov, R. Layton, and I. Gondal, "Optimization based clustering algorithms for authorship analysis of phishing emails," *Neural Process. Lett.*, vol. 46, no. 2, pp. 411–425, Oct. 2017, doi: [10.1007/s11063-017-9593-7](https://doi.org/10.1007/s11063-017-9593-7).
- [37] S. Zawoad, A. K. Dutta, A. Sprague, R. Hasan, J. Britt, and G. Warner, "Phish-net: Investigating phish clusters using drop email addresses," in *Proc. APWG eCrime Researchers Summit*, San Francisco, CA, USA, Sep. 2013, pp. 1–13, doi: [10.1109/eCRS.2013.6805777](https://doi.org/10.1109/eCRS.2013.6805777).
- [38] Y. Han and Y. Shen, "Accurate spear phishing campaign attribution and early detection," in *Proc. 31st Annu. ACM Symp. Appl. Comput.*, Pisa, Italy, Apr. 2016, pp. 2079–2086, doi: [10.1145/2851613.2851801](https://doi.org/10.1145/2851613.2851801).
- [39] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

- [40] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 19–1–19–21, Jul. 2017, doi: [10.1145/3068335](https://doi.org/10.1145/3068335).
- [41] Y. Ren, U. Kamath, C. Domeniconi, and G. Zhang, "Boosted mean shift clustering," in *Proc. Mach. Learn. Knowl. Discovery Databases Eur. Conf. (ECML PKDD)*, in Lecture Notes in Computer Science, vol. 8725. Nancy, France: Springer, 2014, pp. 646–661, doi: [10.1007/978-3-662-44851-9_41](https://doi.org/10.1007/978-3-662-44851-9_41).
- [42] T. Van Craenendonck and H. Blockeel, "Using internal validity measures to compare clustering algorithms," in *Proc. ICML*, vol. 1, no. 1, 2015, pp. 1–8.
- [43] R. Layton, P. Watters, and R. Dazeley, "Unsupervised authorship analysis of phishing webpages," in *Proc. Int. Symp. Commun. Inf. Technol. (ISCIT)*, Gold Coast, QLD, Australia, Oct. 2012, pp. 1104–1109, doi: [10.1109/ISCIT.2012.6380857](https://doi.org/10.1109/ISCIT.2012.6380857).
- [44] A. El Aassal, S. Baki, A. Das, and R. M. Verma, "An in-depth benchmarking and evaluation of phishing detection research for security needs," *IEEE Access*, vol. 8, pp. 22170–22192, 2020, doi: [10.1109/ACCESS.2020.2969780](https://doi.org/10.1109/ACCESS.2020.2969780).
- [45] A. Almomani, B. B. Gupta, S. Atawneh, A. Meulenberg, and E. Almomani, "A survey of phishing email filtering techniques," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 4, pp. 2070–2090, 4th Quart., 2013, doi: [10.1109/SURV.2013.030713.00020](https://doi.org/10.1109/SURV.2013.030713.00020).
- [46] R. Dazeley, J. Yearwood, B. H. Kang, and A. V. Kelarev, "Consensus clustering and supervised classification for profiling phishing emails in internet commerce security," in *Proc. Knowl. Manage. Acquisition Smart Syst. Services, 11th Int. Workshop (PKAW)*, in Lecture Notes in Computer Science, vol. 6232. Daegu, Korea: Springer, 2010, pp. 235–246, doi: [10.1007/978-3-642-15037-1_20](https://doi.org/10.1007/978-3-642-15037-1_20).
- [47] J. Yearwood, D. Webb, L. Ma, P. Vamplew, B. Ofoghi, and A. V. Kelarev, "Applying clustering and ensemble clustering approaches to phishing profiling," in *Proc. 8th Australas. Data Mining Conf. (AusDM)*, vol. 101. Melbourne, VIC, Australia: Australian Computer Society, Dec. 2009, pp. 25–34. [Online]. Available: <http://crpit.scem.westernsydney.edu.au/abstracts/CRPITV101Yearwood.html>
- [48] T. Gangavarapu, C. D. Jaidhar, and B. Chanduka, "Applicability of machine learning in spam and phishing email filtering: Review and approaches," *Artif. Intell. Rev.*, vol. 53, no. 7, pp. 5019–5081, Oct. 2020, doi: [10.1007/s10462-020-09814-9](https://doi.org/10.1007/s10462-020-09814-9).
- [49] A. Karim, S. Azam, B. Shanmugam, K. Kannoorpatti, and M. Alazab, "A comprehensive survey for intelligent spam email detection," *IEEE Access*, vol. 7, pp. 168261–168295, 2019, doi: [10.1109/ACCESS.2019.2954791](https://doi.org/10.1109/ACCESS.2019.2954791).
- [50] S. Kumar BIRTHRIA and A. K. Jain, "A comprehensive survey of phishing email detection and protection techniques," *Inf. Secur. J., A Global Perspective*, vol. 31, no. 4, pp. 411–440, Jul. 2022, doi: [10.1080/19393555.2021.1959678](https://doi.org/10.1080/19393555.2021.1959678).
- [51] K. Althobaiti, N. Meng, and K. Vaniea, "I don't need an expert! Making URL phishing features human comprehensible," in *Proc. CHI Conf. Hum. Factors Comput. Syst.* Yokohama, Japan: ACM, May 2021, pp. 1–17, doi: [10.1145/3411764.3445574](https://doi.org/10.1145/3411764.3445574).
- [52] Cofense PhishMe, "Enterprise phishing resiliency and defense report," PhishMe, Leesburg, VA, USA, Tech. Rep. 3, 2017, Accessed Aug. 2020. [Online]. Available: <https://cofense.com/wp-content/uploads/2017/11/Enterprise-Phishing-Resiliency-and-Defense-Report-2017.pdf>
- [53] D. Ranganayakulu and C. Chellappan, "Detecting malicious URLs in e-mail—An implementation," *AASRI Proc.*, vol. 4, pp. 125–131, Jan. 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2212671613000218>
- [54] L. M. Form, K. L. Chiew, S. N. Sze, and W. K. Tiong, "Phishing email detection technique by using hybrid features," in *Proc. 9th Int. Conf. IT Asia (CITA)*, Sarawak, Kuching, Malaysia, Aug. 2015, pp. 1–5, doi: [10.1109/CITA.2015.7349818](https://doi.org/10.1109/CITA.2015.7349818).
- [55] J.-O. Palacio-Niño and F. Berzal, "Evaluation metrics for unsupervised learning algorithms," 2019, *arXiv:1905.05667*.
- [56] H. Gao, J. Hu, C. Wilson, Z. Li, Y. Chen, and B. Y. Zhao, "Detecting and characterizing social spam campaigns," in *Proc. 10th ACM SIGCOMM Conf. Internet Meas.* New York, NY, USA: Association for Computing Machinery, Nov. 2010, pp. 35–47, doi: [10.1145/1879141.1879147](https://doi.org/10.1145/1879141.1879147).
- [57] R. Layton, P. Watters, and R. Dazeley, "Automatically determining phishing campaigns using the USCAP methodology," in *Proc. eCrime Researchers Summit*, Dallas, TX, USA, Oct. 2010, pp. 1–8, doi: [10.1109/ecrime.2010.5706698](https://doi.org/10.1109/ecrime.2010.5706698).
- [58] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, San Diego, CA, USA, May 2018, pp. 1–12, doi: [10.1109/ECRIME.2018.8376206](https://doi.org/10.1109/ECRIME.2018.8376206).
- [59] I. Fette, N. Sadeh, and A. Tomasic, "Learning to detect phishing emails," in *Proc. 16th Int. Conf. World Wide Web*, Banff, AB, Canada, May 2007, pp. 649–656, doi: [10.1145/1242572.1242660](https://doi.org/10.1145/1242572.1242660).

KHOLOUD ALTHOBAITI received the Ph.D. degree in usable security from the Institute for Language, Cognition and Computation, The University of Edinburgh, in 2020. She is currently an Assistant Professor with the Computer Science Department, Taif University. Her research interests include usable security, such as phishing management and handling and supporting users handling phishing emails.

MARIA K. WOLTERS received the Ph.D. degree in communication research and phonetics from the University of Bonn, Germany, in 2000. She is currently a Reader (an Associate Professor) in design informatics with The University of Edinburgh, Germany, and the Group Leader of the Social Computing Research Group, Research Institute OFFIS, Oldenburg, Germany.

NAWAL ALSUFYANI received the Ph.D. degree in electronics engineering from the University of Kent, U.K., in 2019. She is currently an Assistant Professor in computer science with the College of Computers and Information Technology, Taif University. Her research interests include biometric security, computer vision, pattern recognition, and machine learning algorithms.

KAMI VANIEA received the Ph.D. degree in computer science from Carnegie Mellon University. She is currently a Reader (an Associate Professor) in cyber security with The University of Edinburgh. Her research interest includes the human factors of cyber security and privacy, aiming to better understand the protection needs of all types of users.

...