Beyond Isolated Phishing Emails: Discovering Hidden Campaign Relationships with MCDA

Elyssa Boulila[1,3][0000−0003−4690−7129], Marc Dacier[2][0000−0003−3206−2030], Siva Prem Vengadessa Peroumal[1], Nicolas Veys[1], and Simone Aonzo[3][0000−0001−9547−3502]

[1] Amadeus IT Group, Villeneuve-Loubet, France
{elyssa.boulila,sivaprem.vengadessaperoumal,nicolas.veys}@amadeus.com
[2] KAUST, Thuwal, Kingdom of Saudi Arabia
marc.dacier@kaust.edu.sa
[3] EURECOM, Biot, France
simone.aonzo@eurecom.fr

Abstract. Phishing remains a major threat, with adversaries continuously adapting evasion and anti-analysis techniques. Understanding whether evasive phishing emails are isolated events or manifestations of persistent, evolving activity is an important challenge for defenders. In this work, we do not claim campaign ground truth. Instead, we propose an analyst-oriented clustering methodology based on Multi-Criteria Decision Analysis (MCDA) to surface operationally meaningful relationships among phishing messages that bypassed state-of-the-art protections.
We evaluate the proposed approach on a rare, high-value dataset collected over an 18-month period, comprising user-reported phishing emails that reached end users' inboxes after evading two layers of advanced protection. Our results show that the resulting Multi-Dimensional Clusters are cohesive, stable, and operationally meaningful, and that they capture temporally evolving structures that are substantially fragmented by a density-based clustering algorithm. These relationships often span long time intervals and are supported by subtle but persistent behavioral traces, suggesting that phishing activity evolves gradually through changes in message content, evasion, and anti-analysis techniques.
By enabling longitudinal analysis of stealthy phishing activity, our method helps analysts track evolving related activity over time and generate actionable hypotheses about emerging threats, shared tooling, and persistent attack patterns.

Keywords: Phishing · Threat Attribution. · Threat Intelligence.

1 Introduction

Phishing remains one of the most pervasive threats faced by organizations, with adversaries continuously refining their techniques to evade detection and exploit human vulnerabilities [5]. In this work, we take a closer look at a particularly rare and consequential subset of this ecosystem: stealthy phishing messages that bypass advanced commercial protections and still reach end users' inboxes.

Our objective is not to establish campaign ground truth. Rather, we seek to determine whether these messages are merely unrelated attacks that happened to pass through the "holes in the net," or whether they form persistent and evolving activity patterns whose artifacts gradually change in order to maintain stealth.

Traditional clustering methods struggle in this context because they typically assume stable similarity patterns and consistent feature relevance across samples. These assumptions do not align with the evolving nature of phishing activity, where informative features shift over time and message relationships may persist only through gradual, intermediate changes. Our objective is to identify message groups defined by temporal continuity and partial similarity, even when distant messages within a group no longer share strong direct commonalities.

To address this problem, we rely on Multi-Criteria Decision Analysis (MCDA), and in particular on Ordered Weighted Averaging (OWA), to fuse heterogeneous similarity signals into a single analyst-oriented graph representation. Unlike fixed feature-weighting or distance-averaging schemes, this approach does not assume a single dominant or globally stable feature set. Instead, it allows relationships to emerge from different subsets of indicators across message pairs and across time, while preserving interpretability. This property is especially important in operational settings, where Security Operation Center (SOC) analysts must be able to inspect, understand, and act upon the resulting groupings.

We evaluate our methodology on 18 months of user-reported phishing emails collected across five organizations, comprising one large multinational technology company and four additional entities whose email security is managed by the former. This dataset is intentionally narrow but high-value: it contains credential-harvesting spear-phishing emails that bypassed commercial email protections, were delivered to users, and were subsequently reported and validated. On this dataset, our evidence shows that the resulting clusters are cohesive, stable, and operationally meaningful. Moreover, when compared against a standard density-oriented baseline, the long, temporally chained structures recovered by our approach are substantially fragmented.

Our findings suggest that stealthy phishing activity is often more persistent than individual email observations alone would suggest. At the same time, we stress that the clusters produced by our method should not be interpreted as definitive actor, kit, or campaign labels. Rather, they constitute analyst-facing hypotheses that capture operationally coherent activity patterns, which may reflect evolving campaigns, shared tooling, or coordinated infrastructure reuse.

Our main contributions are:

- An interpretable MCDA-based graph-fusion methodology for correlating phishing messages under temporal drift.
- An empirical analysis of 18 months of user-reported credential-harvesting phishing emails collected from five organizations, providing a rare view into stealthy phishing activity that evaded advanced defenses.
- Evidence that the resulting Multi-Dimensional Clusters are cohesive, stable, and operationally meaningful, despite the absence of campaign ground truth.

– A comparison showing that a standard density-oriented clustering baseline fragments the temporally evolving structures recovered by our approach.
– A case study illustrating the continuous evolution of obfuscation, anti-analysis, and delivery techniques within a long-running phishing activity pattern.

The remainder of this article is structured as follows. Section 2 reviews the state of the art and positions our work within the existing literature. Section 3 describes the dataset used in this study, while Section 4 details the proposed clustering methodology. Section 5 presents the experimental evaluation, followed by Section 6, which examines a representative case study. Finally, we discuss our findings and conclude the paper in Section 7.

2 Related Work

To position our contributions, we review prior work related to phishing clustering and profiling, and we highlight their existing limitations.

Early research on email analysis predominantly approached clustering as a binary classification task, seeking to distinguish malicious emails from benign ones. Within this paradigm, clustering served as a detection mechanism, producing only two categories of interest: legitimate versus malicious. A wide range of machine-learning-based techniques have been investigated in this context (e.g., [3, 17, 22]). In parallel, other studies shifted the focus toward grouping phishing messages into campaign-level clusters [4, 23, 25]. A complementary line of work has investigated clustering as a means to profile phishing kits [12–15, 20, 26, 31]. These efforts generally rely on structural and content-based email features to identify relationships across messages. As a result, they are very good at grouping together messages that all have the same set or subset of features in common but fail to associate them with others that would have evolved over time. Han et al. [16] proposed a graph-based semi-supervised model to attribute spear-phishing messages to known campaigns, though their method relied exclusively on message-level features. More recent studies have broadened the analytical scope by incorporating code- and infrastructure-level artifacts [7, 18, 21, 24, 32]. For example, Lee et al. [18] examined phishing kits through script-level analysis revealing reusable components and shared development practices, while Oest et al. [24] uncovered filtering mechanisms in *.htaccess* files that demonstrate extensive kit reuse. Despite recent advances, the field still lacks formal methods to track the temporal evolution of phishing clusters. As experimentally demonstrated in Section 5, conventional unsupervised clustering is ill-suited to follow evolving campaigns. Existing analyses remain largely manual and expert-driven, limiting scalability and timely attribution.

3 Dataset description

In this study, we select, within our dataset, a specific class of phishing emails, namely the evasive credential-harvesting spear-phishing attempts. Our motivation for selecting this subset of messages arises from their continued prominence as one of the leading initial intrusion vectors in modern cyberattacks

[8]. Furthermore, they are spear-phishing messages, i.e. sent to only a small number of recipients, rather than part of broad campaigns. This makes them, therefore, particularly challenging to analyze, especially when attempting to determine whether seemingly unrelated phishing attempts may, in reality, be associated with long-term, persistent, and evolving campaigns.

To construct our dataset, we proceed in two steps: first, we collect fraudulent messages that evaded email-security filtering; second, we identify those specifically involving credential-harvesting spear-phishing [4]. The resulting dataset serves as input for our MCDA-based clustering methodology.

Collecting evasive fraudulent messages: Our data collection covers an 18-month period from March 2024 to August 2025. We collaborate with five organizations with diverse business activities, collectively employing more than 20,000 collaborators across four continents [5]. We began by gathering all emails reported as *suspicious* by employees within these organizations. These reports represent approximately 0.02% of all delivered emails, amounting to approximately 11,000 user-reported messages per month.

From this initial dataset, we retain only the messages validated as malicious by the SOC responsible for securing the aforementioned organizations. Overall, 3% of all user-reported messages were confirmed to involve fraudulent activities, including phishing, fraud, and invoice-based scams. This percentage corresponds to approximately 6,000 emails obtained over the entire collection period. The remaining messages were flagged as either legitimate (41.3%) or spam (55.7%).

Identifying credential-harvesting spear-phishing: We process the messages obtained from the previous step in real time, immediately after they are reported by users. We rely on *CrawlerBox*, an open-source analysis infrastructure introduced by Boulila et al. [5]. It combines an email parser with a custom web crawler designed to circumvent common client-side cloaking techniques. For instance, it can bypass bot-detection mechanisms relying on browser-fingerprinting strategies. Each reported message is parsed, and all embedded URLs are extracted and crawled. *CrawlerBox* automatically records the network traffic generated during crawling and captures screenshots of the resulting landing pages. To identify credential-harvesting spear-phishing attempts, we compare these screenshots against the legitimate authentication portals used by the organizations under study. Visual similarity is quantified using fuzzy hashing methods, namely pHash and dHash. In this context, two pages are considered similar if their hash-based Hamming distance falls below a manually defined threshold. To mitigate false positives (FPs), all candidate matches are subsequently cross-validated through inspection of the corresponding landing domains. For ambiguous cases, such as those involving legitimate-service abuse (e.g., *.workers.dev, *.pages.dev), we additionally perform manual verification based on the captured screenshots. Using this multi-stage procedure, we confirm the absence of any FPs. Furthermore,

[4] Extending our study to encompass all fraudulent messages is left for future work.

[5] The organizations under study deploy advanced commercial email-security products that filter incoming messages. Only emails categorized as *clean* by these products are delivered to end-users' inboxes.

manual inspection of the pages whose associated Hamming distance falls only slightly below the threshold confirms that no false negatives are identified.

Applying the two steps outlined above results in a final dataset of 721 unique messages. Although modest in size, this unique dataset is very rich and covers a long period of time. It represents an intentionally narrow and analytically valuable subset of attacks, those that (1) successfully bypassed advanced commercial email-security defenses, (2) were detected and reported by end-users, and (3) correspond specifically to credential-harvesting spear-phishing. Because such attacks are both rare and deliberately stealthy, the resulting dataset is inherently limited in volume but exceptionally rich in analytical value.

4 Clustering methodology

4.1 Motivation

To uncover both structural and temporal patterns within the collected dataset, we employ an aggregation approach that contrasts with traditional algorithms such as k-means and HDBSCAN. Although these established methods perform well in static contexts [4, 25], they exhibit notable limitations in dynamic environments, particularly when cluster characteristics evolve over time [23]. As shown in our work (see Section 5), such limitations render conventional clustering techniques inefficient in capturing the temporal variability inherent in phishing campaigns. Moreover, conventional clustering approaches typically assume that features contribute consistently in forming clusters [10]. In practice, it is conjectured that the discriminative power of individual features may vary across clusters or evolve over time. This study examines this conjecture using a clustering approach explicitly designed to reveal such temporal and feature-level evolution.

4.2 Overview

We propose a methodology based on multi-criteria decision analysis (MCDA) which has been used to study other security issues in the past. For instance, spam botnet correlation [29], Rogue AV campaigns [9], and targeted attacks analysis [28], have been analyzed through the lenses of MCDA and have unearthed unknown characteristics of these ecosystems.

As presented in Figure 1 and as detailed below, our methodology integrates three main components:

- **Feature selection:** we first identify a set of relevant features characteristic of phishing attacks, denoted by $F_k, k = 1, \ldots, n$. Subsequently, we extract for each message the corresponding feature values.
- **Graph construction:** for each feature F_k, we define a similarity function s_k that measures how similar two messages are with respect to that specific feature. For example, when comparing message bodies treated as feature values, we use the *cosine similarity* of their text embeddings. Using these similarity values, we construct an undirected, edge-weighted graph G_k where nodes V_k represent messages with respect to feature F_k, and edges E_k encode pairwise similarities $s_k(i, j)$ between message representations i and j.

– **Multi-criteria data fusion:** The individual weighted graphs G_k are fused into one single aggregated graph G_* using an aggregation function designed to reflect relevant expert knowledge. To ground the definition of this function in operational practice, we designed and piloted our experiment in collaboration with a Subject-Matter Experts (SME) team comprising four senior incident-response practitioners handling phishing case investigations. Crucially, the SME team did not participate in the definition, design, or implementation of our methodology. Their role was limited to sharing domain expertise based on their operational experience.

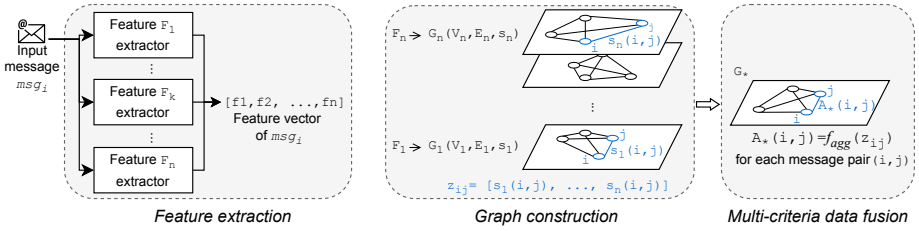


Fig. 1: A representation of our MCDA-based correlation approach. Multiple edge-weighted graphs are fused using an aggregation function f_{agg}.

We provide a more detailed description of our approach below. In particular, we present our chosen aggregation function and explain how clusters are subsequently derived from the aggregated graph G_*.

4.3 Feature selection

We select relevant features derived from both message-level analysis and the automated crawling of embedded web resources. The similarity metrics applied to these features are detailed in Section 4.4.

Static message-level features. We select a set of features that we consider likely to reveal correlations among individual messages and enable their aggregation into campaigns. These include the message reception date (F_{date}), which records the timestamp at which the message was received; the sender's IP address aggregated to its /16 subnet ($F_{senderIP}$); and the content of the message body (F_{msg_body}).

Dynamic features – obtained from crawling embedded web resources. *CrawlerBox* processes each message by parsing its content and extracting embedded web resources, such as URLs and HTML attachments. These resources are then rendered within an automated browser instance [5]. We derive several feature sets from the resulting network traffic:

1. Landing domain ($F_{landing\_domain}$): we extract the second-level domain (SLD) using the Public Suffix List [2].
2. Loaded domains ($F_{loaded\_domains}$): during page rendering, additional domains may be requested (e.g., for scripts, images, fonts, or as part of redirection chains). We collect these domains, strip subdomains above the SLD, and treat

the resulting set as a feature. This captures the supporting infrastructure of the phishing campaign, which often spans multiple related domains.

3. URL path segments ($F_{first\_segments}$ and $F_{last\_segments}$): for each loaded URL, we extract the first and last path segments, defined as the first and last portions of the URL path separated by slashes (/). These segments often encode campaign identifiers or payload delivery paths, making them valuable for clustering and attribution.
4. URL query parameters ($F_{query\_parameters}$): these parameters can be indicative of specific phishing kits as described in previous reports [1].
5. POST requests parameters ($F_{POST\_parameters}$): we extract the parameters from POST requests if any. POST data often carries structured information containing exfiltrated data such as credentials, tokens, or browser fingerprints. Collecting these parameters can reveal the attacker's data collection strategy and provide strong attribution signals.
6. Loaded images (F_{loaded_images}): we collect all images rendered during page loading and compute their perceptual hashes (pHash).
7. JavaScript identifiers and strings ($F_{JS\_identifiers}$ and $F_{JS\_strings}$): we analyze JavaScript code loaded during page rendering by filtering network traffic to retain only scripts originating from the primary landing domain. From the retained scripts, we extract the set of identifiers (excluding declared variable and function names) and string literals.
8. Responses returned as a result of XMLHttpRequest (XHR) requests ($F_{XHR\_responses}$): some phishing websites leverage APIs such as XMLHttpRequest to transmit fingerprinting data to the attacker's server [19]. Consequently, we incorporate XHR responses as a feature in our analysis.
9. Four additional features are obtained from the Document Object Model (DOM): first, we extract the content of CSS qualified rules and at-rules, denoted as $F_{CSS\_qualified\_rules}$ and $F_{CSS\_at\_rules}$ [6]. Next, we remove comments and hidden elements (e.g., text with extremely small font sizes, zero opacity, transparent color, or zero dimensions). This preprocessing ensures that only visible content is analyzed. Next, we extract the following features: the DOM structure ($F_{DOM\_structure}$) and the visible text within the page (F_{DOM_text}).

By systematically extracting these features, we aim to capture both static and dynamic indicators of phishing activity, enabling robust attribution.

Note that some features may be partially correlated, for example $F_{DOM\_structure}$ and F_{DOM_text}. In our case, redundancy is beneficial as it improves the resilience of our approach under noisy or adversarially manipulated observations. In practice, attackers often perturb a single feature family at a time to circumvent detection (e.g., by randomizing identifiers or slightly altering the DOM structure). However, they rarely modify all feature families consistently without incurring additional development effort or breaking functionality. As a result, retaining multiple partially overlapping features increases resilience to evasion, since similarity can still be supported by the remaining, less perturbed

[6] Qualified rules specify style declarations applied to document elements, whereas at-rules define special processing rules or values (e.g., @import, @media).

indicators. Finally, to ensure that the correlated features do not dominate the decision unintentionally, we assess the stability of the obtained clusters and the contribution of these features to the final outcome. In particular, a sensitivity analysis is performed in Section 4.6.

4.4 Graph construction

We use five different similarity metrics according to the feature type:

1. the *equality* for exact values such as $F_{senderIP}$ and $F_{landing\_domain}$
2. the *Jaccard* coefficient for the features representing sets of values, i.e., $F_{loaded\_domains}$, $F_{first\_segments}$, $F_{last\_segments}$, $F_{query\_parameters}$, $F_{POST\_parameters}$, F_{loaded_images}, $F_{JS\_identifiers}$, $F_{JS\_strings}$), $F_{XHR\_responses}$, $F_{CSS\_qualified\_rules}$ and $F_{CSS\_at\_rules}$.
   For F_{loaded_images}, we compute the Jaccard coefficient over sets of fuzzy hashes. In this case, the standard definitions of set intersection and union are adapted by replacing exact equality with a similarity-based equivalence. Specifically, two hashes are treated as identical whenever their normalized Hamming distance is below 0.1. We further explain this specific implementation in Appendix A.
3. a custom function that maps date differences expressed in days (F_{date}) to the interval [0, 1], according to various linear segments defined by parameters a, b, c:
$$f(d_{ij}, a, b, c) = \begin{cases} 1, & \text{if } d_{ij} \leq a, \\ 1 - \dfrac{d_{ij} - a}{2(b-a)}, & \text{if } a \leq d_{ij} \leq b, \\ \dfrac{1}{2} - \dfrac{d_{ij} - b}{2(c-b)}, & \text{if } b \leq d_{ij} \leq c, \\ 0, & \text{if } d_{ij} \geq c. \end{cases}$$
   where d_{ij} is the date difference in days between two messages i, j. For this analysis, the parameters a, b, c have been set to 15, 30, 45, respectively.
4. the *cosine similarity* applied to text embeddings obtained with Sentence-BERT for text-based features, i.e., F_{msg_body} and F_{DOM_text}.
5. the *sequence comparison* of HTML tags applied to $F_{DOM\_structure}$: we iterate over the parsed DOM in document order, and we extract a linear sequence of tag names. We compare the obtained sequences using the Ratcliff–Obershelp similarity algorithm [11]. The resulting score (0–1) measures the documents' structural similarity. We acknowledge that linearizing the HTML DOM into a sequence of tag names abstracts away some hierarchical structural information. However, this abstraction increases robustness to minor structural noise that could be adversarially included by attackers to bypass detection. Additionally, any loss of structural detail is mitigated by the multi-feature design of our approach, where DOM-based similarity is combined with complementary static and dynamic features in the fused graph.

Using these metrics, we obtain 17 similarity matrices $A_k, k = 1, \ldots, 17$, representing the similarities between pairs of messages in each feature space F_k.

4.5 Multi-criteria data fusion

For each message pair (i, j), we construct a similarity vector

$$\mathbf{z_{ij}} = [A_1(i,j), A_2(i,j), \ldots, A_{17}(i,j)]$$

where each component $z_{ij}(k) = A_k(i,j) = s_k(i,j) \in [0,1]$ denotes the similarity between the two messages i, j with respect to feature F_k, given $k = 1, \ldots, 17$.

In order to group messages into meaningful clusters, we need a single aggregated similarity score per pair, as opposed to a 17-dimensional vector. The choice of an aggregation function in our context is very important. First, it should reflect that not all features contribute equally to detecting correlations. But most importantly, it should reflect that feature relevance may differ across clusters and evolve over time. To address this, we use *Ordered Weighted Averaging* (OWA [30]). This technique enables rank-based aggregation rather than feature-based weighting. It is defined as follows:

For a given weighting vector $\mathbf{w}$*,* $w_i \geq 0$*,* $\sum w_i = 1$*, the* OWA *aggregation function is defined by:*

$$OWA_{\mathbf{w}}(\mathbf{z}) = \sum_{k=1}^{n} w_{(k)} z_{\downarrow(k)}$$

where the notation $\mathbf{z}_{\downarrow}$ *represents the vector obtained from* z *by arranging its components in decreasing order:* $z_{(1)} \geq z_{(2)} \geq \cdots \geq z_{(n)}$

The OWA technique first arranges the components of $\mathbf{z}$ in decreasing order. Next, it multiplies it by a weighting vector $\mathbf{w}$ to obtain a final value. This technique ensures that we associate weights to *magnitude* of values, rather than to specific features. For instance, a weight vector $[1, 0, \ldots, 0]$ reduces the operator to the maximum function, emphasizing the strongest evidence; $[0, \ldots, 0, 1]$ corresponds to the minimum, requiring consensus across all features; and a uniform vector $[\frac{1}{n}, \ldots, \frac{1}{n}]$ yields the arithmetic mean, treating all similarities equally. This flexibility allows the aggregation to reflect domain-specific preferences, balancing optimism, pessimism, or compensatory behavior in the clustering process.

Unlike feature-weighted similarity or metric learning approaches, OWA assigns weights to the ranked similarity values rather than to specific features. This allows correlations to emerge from different subsets of features across message pairs and clusters, without assuming global or stable feature importance. In contrast to averaging-based fusion, strong evidence in a limited number of dimensions is preserved even when other features diverge, and this is essential for tracking evolving phishing campaigns exhibiting partial and non-uniform drift. Moreover, for correlating phishing campaigns over extended periods of time, we need a weighting vector that gives strong but gradually decreasing importance to the highest-ranked values, while it prevents weak or noisy evidence from dominating the aggregation. To achieve this, we rely on a weighting vector $\mathbf{w}$ with a hybrid profile: a linear decay for the first j positions (i.e., the top-ranked values), followed by an exponential decay for the remaining $n - j$ positions (the tail).

In this study, we set $j = 5$, which corresponds to the minimum number of high-similarity feature families empirically observed to indicate shared campaign provenance. This value aligns with the structure of our feature set (content, infrastructure, visual, script, and DOM-level signals) and was validated through pilot analysis with SMEs, who confirmed that the presence of five high similarity scores constitutes a strong indicator that two messages are related and are unlikely to be grouped merely by chance. Emphasizing these top five components ensures that the most reliable evidence of correlation drives the aggregation, while the remaining similarities, though potentially informative, have progressively less impact. To corroborate this expert-driven intuition, we conduct a sensitivity analysis of the parameter j in Section 4.6. Finally, the weighting vector is normalized such that $\sum w_i = 1$.

Next, we aggregate the set of similarity matrices A_k into a single matrix A_*, where each entry represents the OWA-based aggregated similarity score between a pair of messages; that is, $A_*(i, j) = OWA_w(z_{ij})$ for every message pair (i, j). Based on A_*, we construct an undirected graph G_* whose nodes correspond to the messages in the dataset and whose edges encode the OWA similarity scores.

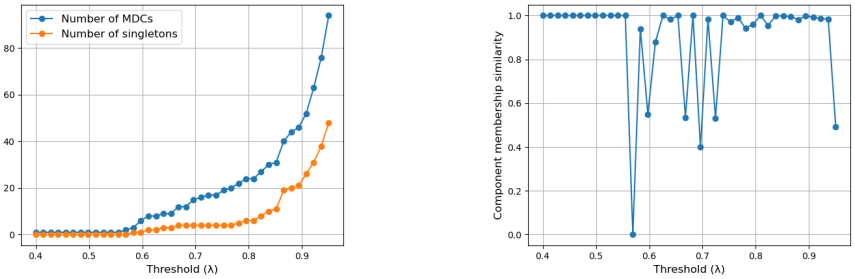
To ensure that the graph captures only meaningful associations, an edge between two nodes is retained only if its value exceeds a decision threshold λ. A sensitivity analysis for selecting an appropriate λ value is presented in Section 5. The connected components of G_* constitute the resulting clusters, which we refer to as *Multi-Dimensional Clusters (MDCs)*. An MDC may comprise subsets of messages characterized by distinct feature combinations; however, all messages within the same MDC are still directly or indirectly connected by at least a sufficient number of features. This approach enables us to track the evolution of clusters in time when features' relevance might evolve. In particular, it enables the identification of persistent stealthy campaigns within the dataset.

4.6 Sensitivity analysis

The decision threshold λ directly shapes the resulting Multi-Dimensional Clusters (MDCs). Low values lead to overly large clusters, while high values fragment the dataset into many singletons. If λ is set too high, messages belonging to successive versions of the same campaign may be split apart, breaking the temporal continuity we aim to preserve. An appropriate λ is therefore one that yields a stable and interpretable partition, broad enough to capture meaningful relationships yet selective enough to avoid artificial merging. To identify such a value, we evaluate the stability of the structure across successive thresholds through component membership similarity: we quantify how consistent the assignment of nodes to MDCs is when slightly increasing the threshold value (i.e., when incrementally removing edges). If the component memberships vary only marginally between two nearby thresholds, this indicates that the resulting partition is robust and that the separation between clusters is stable. To quantify this stability, we compute the Adjusted Rand Index (ARI) [27] between partitions obtained at successive threshold values.We choose the ARI metric for assessing

cluster stability as it is sensitive to membership changes, penalizing both splits and merges of clusters while correcting for chance agreement.

Figure 2a reports the number of MDCs obtained for each λ as well as the number of singletons, while Figure 2b displays the corresponding ARI values for successive threshold values. For thresholds $\lambda \leq 0.57$, the entire dataset collapses into a single MDC, reflecting a fully aggregated structure. As λ increases beyond 0.57, this unified cluster progressively fragments, producing a series of unstable partitions as new clusters begin to emerge. A stability plateau emerges for thresholds in the range $0.82 \leq \lambda \leq 0.93$, where the ARI values stabilize around 1, indicating that successive partitions become nearly identical. However, beyond $\lambda = 0.82$, the number of MDCs increases sharply. This increase is primarily due to the creation of additional singleton clusters rather than substantial changes to the core partition structure. Consequently, the ARI remains close to 1 despite the apparent growth in the number of MDCs.



(a) Number of obtained MDCs (b) ARI score

Fig. 2: Sensitivity analysis of the decision threshold (λ).

To balance cluster stability, interpretability, and robustness to gradual changes in feature relevance over time, we select $\lambda = 0.82$ as an appropriate threshold. This results in partitioning our dataset into 27 distinct MDCs of which 20 have more than one element. Thus, only 8 messages out of 721 form singleton clusters.

It is worth emphasizing that we do not claim that our choice of λ generalizes across datasets. Rather, the contribution lies in the ARI-based stability analysis, which provides a principled, dataset-specific procedure for selecting λ in the presence of temporal drift and chain-like cluster structures. Regarding its generalization, this remains a promising direction for future work. Further investigation is needed to understand whether λ varies substantially across datasets with different characteristics or whether a more universal value could consistently yield meaningful results.

Concerning the value of the parameter j, it was initially set to five based on pilot analyses with subject-matter experts (SMEs). To further validate this choice, we conduct a sensitivity analysis by varying j from 4 to 6 and examining the resulting cluster configurations. For each $j \in 4, 5, 6$, we fix j and perform an ARI-based stability analysis with respect to λ (as explained above) to identify a

value of λ yielding stable clusters. This procedure results in 19 MDCs for $j = 4$ and $\lambda = 0.79$, 27 MDCs for $j = 5$ and $\lambda = 0.82$ (the SME-selected configuration), and 18 MDCs for $j = 6$ and $\lambda = 0.68$. The lower number of MDCs for $j = 4$ and $j = 6$ indicates greater aggregation compared to $j = 5$.

We retain $j = 5$ as it provides a favorable balance between cluster granularity and aggregation, yielding a differentiated MDC structure without compromising stability. In addition, this configuration aligns with expert judgment. In the next sections, we conduct both quantitative and qualitative analyses of the clusters obtained with $j = 5$. As will be presented, the results confirm the consistency and coherence of the resulting MDCs.

5 Evaluation of Multi-Dimensional Clusters

The goal of this evaluation is not to recover a "true" clustering, which is ill-defined in evolving adversarial settings, but rather to assess whether the induced structure is cohesive, stable, and operationally meaningful.

5.1 Quantitative analysis

In our setting, clusters are not expected to be compact or spherical, as campaign evolution induces elongated, temporally chained structures. Consequently, internal validation metrics emphasizing connectivity and cohesion are more appropriate than compactness-based measures. Our evaluation therefore focuses on intra-cluster edge strength and structural stability rather than separation alone.

Since no ground-truth labels are available for our dataset, conducting an external validation of the clustering results is not feasible. Consequently, the evaluation relies exclusively on internal validation methods, which assess the quality of the obtained partitions based solely on intrinsic properties of the data and the induced cluster structure. Additionally, clusters in our context are not necessarily expected to be compact, as they may represent phishing campaigns that evolve gradually over time. This temporal drift can produce elongated or sparse cluster structures. Consequently, traditional clustering evaluation metrics may be inadequate, and metrics that account for structural evolution are more appropriate. Therefore, to conduct a meaningful assessment of our results, we rely on *cluster cohesion* as an internal validation method. For each identified cluster, we measure both the median and mean of the intra-cluster edge weights. The median edge weight provides a robust estimate of typical connection strength within the cluster, minimizing the influence of extreme values. The mean edge weight, in turn, offers a global summary of the cluster's internal connectivity, reflecting overall link strength. Although the mean is more sensitive to outliers, it complements the median by capturing cumulative cluster cohesion.

Figure 3 shows the cohesion and size of each resulting MDC. Both median and mean intra-cluster edge weights consistently exceed the decision threshold $\lambda = 0.82$ in all multi-node clusters (excluding `MDC19` which only comprises two messages). This indicates that the clusters are not only well-connected but also internally consistent, as most pairwise correlations remain substantially above the minimum required for forming a meaningful MDC. Additionally, we

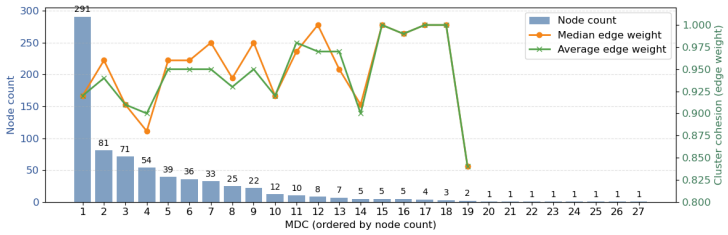


Fig. 3: Quantitative evaluation of the obtained multi-dimensional clusters.

obtain eight singleton clusters, which likely correspond to isolated attacks. These singletons may also correspond to testing/probing activities preceding a broader campaign. They can also arise from highly aggressive polymorphism that prevents the formation of edges above λ. In our dataset, these singleton messages are distributed across different days throughout the collection period, which further suggests sporadic activity rather than a single concentrated burst.

We provide in Section 5.2 a detailed examination of the resulting MDCs. We also compare our approach with standard unsupervised clustering techniques and highlight the limitations of these methods in capturing the multi-dimensional correlations revealed by our data-fusion process.

5.2 Qualitative analysis

As detailed in Section 3, our study relies on a rich corpus of user-reported phishing emails, comprising 721 distinct stealthy messages collected over an 18-month period. Using our clustering approach relying on data fusion and multi-criteria decision analysis, we identify 27 Multi-Dimensional Clusters.

We introduce in this section the main insights derived from our analysis.

Finding 1: *Feature contribution varies across clusters.* Figure 4 reports feature-wise compactness of the obtained MDCs. Several clusters exhibit notably high overall compactness, most prominently `MDC12`, `MDC17`, and `MDC18`. These clusters draw substantial and relatively balanced contributions from a wide range of features. In contrast, the earliest clusters in the ordering (e.g., `MDC1-5`) display markedly lower compactness, suggesting greater intra-cluster variation and evolving attributes. Importantly, no single feature exhibits uniformly high compactness across all clusters, confirming that campaign similarity is multi-dimensional and not driven by a dominant signal. Nevertheless, we observe that several features consistently co-occur with high compactness across a few MDCs, mainly the landing page DOM structure ($F_{DOM\_structure}$), visible text (F_{DOM_text}), JavaScript identifiers ($F_{JS\_identifiers}$), and loaded images (F_{loaded_images}).

Finding 2: *Most messages pertain to persistent, long-term campaigns, as opposed to one-shot isolated attacks.* We offer a graphical representation of the temporal distribution of the messages composing each MDC in Figure 7 in Appendix B. Suffice it to say that each of the ten largest clusters exists for several months. Collectively, they contain 664 messages, accounting for 92.1% of the dataset. In particular, `MDC1` and `MDC4` exhibit the longest activity windows,

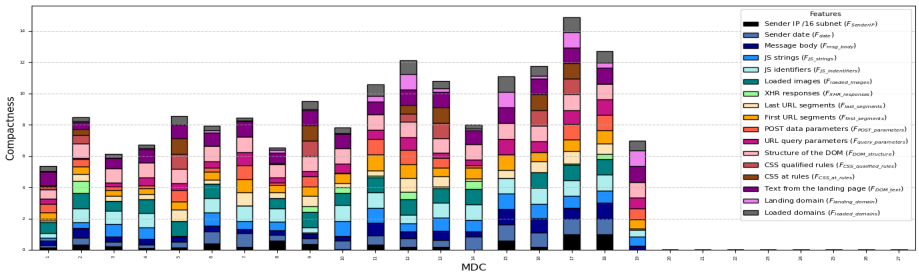


Fig. 4: Evaluation of feature contribution across MDCs.

spanning 460 and 481 days respectively, and thus covering the majority of our analysis period. This indicates that the adversaries behind these clusters operated continuously over extended periods rather than launching short-lived or opportunistic attacks. The persistence and regular reappearance of samples within these MDCs strongly suggest structured, sustained campaigns maintained over more than a year.

Finding 3: *Traditional unsupervised clustering algorithms fail to identify evolving clusters.* We compare our results with those obtained using HDBSCAN, a hierarchical variant of the density-based DBSCAN clustering. We select this technique as it has been utilized in prior studies on phishing clustering [23], and because it operates without requiring hyper-parameter tuning or prior knowledge of the number of clusters. We provide as input a pairwise distance matrix obtained by averaging the distance matrices $D_k = 1 - A_k$, where A_k denotes the pairwise similarity matrix associated with feature F_k for k=1,...,17. HDBSCAN labels 141 out of 721 samples as noise and identifies 87 clusters. Notably, it heavily fragments large MDCs: for instance, `MDC1` is split into 32 clusters. This result highlights the inability of HDBSCAN, when combined with naïve fusion strategies, to maintain cluster continuity when attacks evolve over time. In particular, we describe in Section 6 the temporal evolution of `MDC1`. We acknowledge that applying HDBSCAN with averaged distance aggregation may limit its ability to capture chain-like structures. In the phishing domain, however, there is no established or canonical strategy for fusing heterogeneous signals, and simple averaging remains an acceptable choice due to its minimal hyperparameter requirements. While a carefully engineered or learned fusion mechanism could potentially improve HDBSCAN's performance, developing such an approach would significantly broaden the scope of this work and introduce additional design choices that increase the risk of overfitting.

6 Case study

We present in this section the characteristics of `MDC1` and the evolution of its attributes in time. This cluster was not fully caught using traditional clustering algorithms. It is the largest cluster identified, and it extends over a substantial period of time, spanning 460 days.

Figure 5 displays the OWA-aggregated pairwise similarity matrix for the samples belonging to `MDC1`, ordered chronologically by message delivery date (from left to right and top to bottom). Each cell represents the fused similarity between two messages, obtained by aggregating their multi-dimensional feature similarities using the OWA operator. The heatmap reveals a clear temporal structure: large yellow blocks along the diagonal correspond to subsets of messages that are both highly similar and temporally adjacent. The gradual shift from yellow to green or blue further indicates the progressive evolution of the underlying campaign, as its artifacts and techniques change over time. Notably, only limited similarity persists between the oldest and most recent messages, reflecting a temporal drift. Specifically, the first and last messages within `MDC1` exhibit an aggregated similarity score of 0.39, which falls below the decision threshold $\lambda = 0.82$. Although these messages lack strong direct feature-level similarities, they remain connected through robust intermediary links.

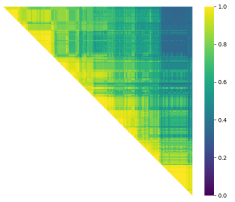


Fig. 5: OWA-aggregated similarity heatmap for samples within `MDC1`.

We uncover next the characteristics of this cluster.

Finding 4: *Obfuscation, dynamic rendering, and bot detection are prevalent within long-lasting campaigns.* The obtained Document Object Models (DOMs) have a few attributes in common. First, they exhibit a consistent pattern in the semantic naming of variables, functions, and DOM sections. Second, all of them rely on multi-layer obfuscation techniques, typically combining encoded payloads with runtime evaluation. This enables the malicious content to be constructed dynamically in the browser while remaining concealed from static inspection.

Third, the pages are rendered through client-side scripts that progressively build DOM elements, including the injection of branded login flows, timed transitions, and auto-filling fields. This dynamic rendering prevents structural signatures from being visible in the initial HTML response. Additionally, all attacks employ bot-detection and anti-analysis mechanisms as cloaking techniques, either by abusing legitimate services such as Cloudflare's Turnstile or by implementing custom client-side checks. Finally, we observe a common pattern of asynchronous, token-based communication with a controller, often using encrypted payloads. This indicates a coordinated effort to evade detection and support real-time orchestration of the phishing workflow.

Finding 5: *Evasion techniques exhibit a rapid and continuous evolution.* As explained above, the collected DOMs share the same phishing logic and semantic naming of variables and functions. Nonetheless, they exhibit progressively evolving capabilities, likely reflecting continuous adaptation to email-security countermeasures. To capture this evolution, we process the DOMs chronologically and represent each page through its set of extracted JavaScript identifiers. We construct "DOM versions" by comparing each new DOM with the versions previously identified. Similarity to existing versions is quantified using the Jaccard

coefficient. When the similarity to any existing version exceeds 90%, the page is assigned to that version; otherwise, a new version is identified, and the page is used as its reference. The result is a sequence of versions that capture structural changes over time, as illustrated in Figure 6.

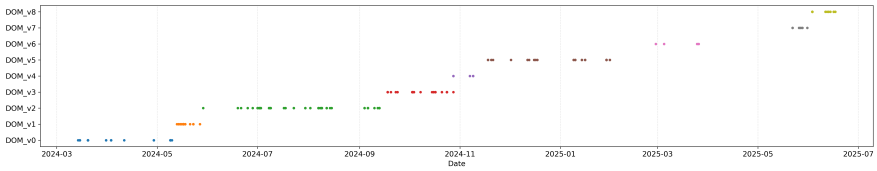


Fig. 6: DOM versions as extracted from `MDC1`.

We obtain 9 distinct DOM versions. We subsequently compare their corresponding references. Mainly, we notice an increasing obfuscation effort. In the earliest versions, the content of the DOM is concealed using a simple XOR-based encoding function, indicating an initial attempt to hinder static analysis. A later version introduces a layered decoding process in which the payload is decrypted using a Caesar cipher, then Base64-decoded and executed via `eval`. In the most recent versions, the DOM is AES decrypted and executed via obfuscated `eval` functions, thereby significantly complicating both static and dynamic analysis. Concerning anti-analysis techniques, we observe that the later versions incorporate multiple defensive checks, including the detection of instrumented browsers, the interception of developer-tool shortcuts, the disabling of the context menu, and timing-based debugger detection.

We emphasize that MDCs should not be interpreted as ground-truth actor or phishing-kit groupings. Instead, they represent analyst-facing abstractions that capture operationally coherent activity patterns, which may correspond to campaigns, shared tooling, or coordinated infrastructure reuse.

Finding 6: *External threat-intelligence indicators corroborate the attribution of `MDC1` to Tycoon2FA.* To attribute this cluster to known threat actors, we rely on external threat intelligence indicators. In particular, we incorporate URL patterns associated with various Adversary-in-the-Middle phishing threats, as documented in a recent report [6]. These patterns enable us to label 347 out of 721 samples (48.1% of the dataset). We do not use these labels as input for the evaluation phase, as they do not cover the entire dataset and may be overly generic (for instance, domain names matching the pattern `[a-z0-9]{2,6}.[a-z]{5,15}.(ru|es|cc|info|su|vip)`). Nevertheless, these labels remain valuable for post-hoc analysis. Within this cluster, 155 out of 291 messages are labeled, all of which are associated with Tycoon2FA, a known Phishing-as-a-Service platform. To further assess potential overlaps, we examine whether samples from other MDCs match the URL patterns commonly associated with this platform, identifying 36 such matches. However, manual inspection of the corresponding DOM structures and loaded scripts reveals that the semantics and overall structure of these attacks differ entirely from those observed in `MDC1`. These matches arise from overly generic domain patterns, for example, `[a-z0-9]{2,6}.[a-z]{5,15}.es`,

which also matches benign domains such as *www.google.es*. Consequently, the 36 cases represent false attributions caused by non-discriminative URL pattern matching. This observation highlights several important lessons. First, even partially imprecise threat intelligence can be effective in characterizing clusters at a higher level. However, such indicators are too generic to serve as reliable inputs for clustering, as they may introduce attribution errors. Second, the strong concentration of Tycoon2FA-related indicators within this cluster suggests that the clustering approach successfully groups messages sharing common provenance, despite variations in content or obfuscation techniques. Finally, this result underscores the complementarity between unsupervised clustering and external intelligence: while threat intelligence alone is insufficient for comprehensive labeling, it can meaningfully enhance interpretability and support post-hoc attribution when combined with data-driven methods.

7 Discussion and Conclusion

We construct multiple graphs capturing pairwise message correlations within individual feature spaces and a fused graph that represents their combined relationships. The fused graph helps analysts explain why two samples are related by exposing intermediate nodes and links, while the feature-specific graphs provide deeper insight into the individual signals driving the overall correlation. SMEs validated our findings, confirming that the MCDA-based clustering reveals relationships between phishing messages that had remained invisible to them so far. The analysts are currently integrating our analytical outputs into their investigation process to enrich incident observables.

Our study nonetheless presents the following limitations. The absence of ground-truth campaign labels prevents direct external validation of the clustering results. However, multiple independent signals suggest that the discovered clusters capture meaningful campaign-level activities. First, clusters exhibit strong temporal coherence, with several spanning hundreds of days. Second, messages within clusters share evolving but structurally related artifacts, such as progressively modified DOM structures and obfuscation techniques. Third, external threat-intelligence indicators independently associate substantial subsets of our dataset with known phishing-as-a-service platforms. Furthermore, we acknowledge that user-reported data is inherently biased and may miss certain campaigns. Additionally, the deployed commercial filters may introduce bias in the set of messages that evade detection. Despite these limitations, applying our methodology to this dataset enables the analysis of temporal shifts in adversary behavior. Our goal is not to perform high-throughput campaign labeling at global email scale, but rather to support analyst-driven hypothesis generation and facilitate deeper understanding of the evasive phishing ecosystem. Finally, evaluating the scalability of our approach on larger and more heterogeneous datasets remains an important direction for future work.

We emphasize that MDCs should not be interpreted as ground-truth actor or phishing-kit attribution. Instead, they capture persistent activity patterns that may correspond to campaigns, shared tooling, or coordinated infrastructure reuse.

Ultimately, the ability to uncover these long-lived and evolving relationships suggests that stealthy phishing activity is far more structured and persistent than individual email observations alone would suggest.

Acknowledgments. This work benefited from two government grants managed by the French National Research Agency with references: "ANR-22-PECY-0007" and "ANR-23-IAS4-0001".

References

1. Mamba 2fa: A new contender in the aitm phishing ecosystem. https://blog.sekoia.io/mamba-2fa-a-new-contender-in-the-aitm-phishing-ecosystem/ (2024)
2. Public suffix list. https://publicsuffix.org/ (2025)
3. Al-Sabbagh, A., Hamze, K., Khan, S., Elkhodr, M.: An enhanced k-means clustering algorithm for phishing attack detections. Electronics **13**(18), 3677 (2024). https://doi.org/10.3390/electronics13183677
4. Althobaiti, K., Wolters, M.K., Alsufyani, N., Vaniea, K.: Using clustering algorithms to automatically identify phishing campaigns. IEEE Access **11**, 96502–96513 (2023). https://doi.org/10.1109/ACCESS.2023.3310810
5. Boulila, E., Dacier, M., Peroumal, S.P.V., Veys, N., Aonzo, S.: A closer look at modern evasive phishing emails. In: 2025 55th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN). pp. 650–663 (2025). https://doi.org/10.1109/DSN64029.2025.00066
6. Bourgue, Q., Clermont, G., TDR team: Global analysis of adversary-in-the-middle phishing threats. https://t7f4e9n3.delivery.rocketcdn.me/wp-content/uploads/2025/06/Sekoia_io___Global_analysis_of_Adversary_in_the_Middle_phishing_threats.pdf (June 2025), [Accessed 03-12-2025]
7. Castano, F., Fernandez, E.F., Alaiz-Rodríguez, R., Alegre, E.: Phikita: Phishing kit attacks dataset for phishing websites identification. IEEE Access **11**, 40779–40789 (2023). https://doi.org/10.1109/ACCESS.2023.3268027
8. Clay, J.: Complete guide to protecting seven attack vectors. https://www.trendmicro.com/en_us/research/22/k/cyber-attack-vectors-how-to-protect-them.html (August 2024), [Accessed 03-12-2025]
9. Cova, M., Leita, C., Thonnard, O., Keromytis, A.D., Dacier, M.: An analysis of rogue av campaigns. In: International Workshop on Recent Advances in Intrusion Detection. pp. 442–463. Springer (2010). https://doi.org/10.1007/978-3-642-15512-3_23
10. De Amorim, R.C.: A survey on feature weighting based k-means algorithms. Journal of Classification **33**(2), 210–242 (2016). https://doi.org/10.1007/s00357-016-9208-4
11. Elmobark, N.: A comparative analysis of python text matching libraries: A multilingual evaluation of capabilities, performance and resource utilization. International Journal of Environment, Engineering and Education **7**(1), 48–60 (2025). https://doi.org/10.55151/ijeedu.v7i1.188
12. Favoretti, J.P., Dantas, F., Pereira Jr, L.A.: Inside the phishing reel: Leveraging browser instrumentation to analyse evasive phishing. In: Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg). pp. 546–562. SBC (2025)
13. Hamid, I.R.A., Abawajy, J.H.: Profiling phishing email based on clustering approach. In: 2013 12th IEEE International Conference on Trust, Security and Privacy in Computing and Communications. pp. 628–635. IEEE (2013). https://doi.org/10.1109/TrustCom.2013.76

14. Hamid, I.R.A., Abawajy, J.H.: An approach for profiling phishing activities. Computers & Security **45**, 27–41 (2014). https://doi.org/10.1016/j.cose.2014.04.002
15. Hamid, I.R.A., Samsudin, N.A., Mustapha, A., Arbaiy, N.: Dynamic trackback strategy for email-born phishing using maximum dependency algorithm (mda). In: International Conference on Soft Computing and Data Mining. pp. 263–273. Springer (2016). https://doi.org/10.1007/978-3-319-51281-5_27
16. Han, Y., Shen, Y.: Accurate spear phishing campaign attribution and early detection. In: Proceedings of the 31st Annual ACM Symposium on Applied Computing. pp. 2079–2086 (2016). https://doi.org/10.1145/2851613.2851801
17. Karim, A., Azam, S., Shanmugam, B., Kannoorpatti, K.: Efficient clustering of emails into spam and ham: The foundational study of a comprehensive unsupervised framework. IEEE access **8**, 154759–154788 (2020). https://doi.org/10.1109/ACCESS.2020.3017082
18. Lee, W., Hur, J., Kim, D.: Beneath the phishing scripts: A script-level analysis of phishing kits and their impact on real-world phishing websites. In: Proceedings of the 19th ACM Asia Conference on Computer and Communications Security. pp. 856–872 (2024). https://doi.org/10.1145/3634737.3657013
19. Lin, X., Ilia, P., Solanki, S., Polakis, J.: Phish in sheep's clothing: Exploring the authentication pitfalls of browser fingerprinting. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1651–1668. USENIX Association, Boston, MA (Aug 2022), https://www.usenix.org/conference/usenixsecurity22/presentation/lin-xu
20. Ma, L., Yearwood, J., Watters, P.: Establishing phishing provenance using orthographic features. In: 2009 eCrime Researchers Summit. pp. 1–10. IEEE (2009). https://doi.org/10.1109/ECRIME.2009.5342604
21. Merlo, E., Margier, M., Jourdan, G.V., Onut, I.V.: Phishing kits source code similarity distribution: A case study. In: 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER). pp. 983–994. IEEE (2022). https://doi.org/10.1109/SANER53432.2022.00116
22. Mondal, S., Maheshwari, D., Pai, N., Biwalkar, A.: A review on detecting phishing urls using clustering algorithms. In: 2019 International Conference on Advances in Computing, Communication and Control (ICAC3). pp. 1–6. IEEE (2019). https://doi.org/10.1109/ICAC347590.2019.9036837
23. Nahapetyan, A., Khare, K., Schwarz, K., Reaves, B., Kapravelos, A.: Characterizing phishing pages by javascript capabilities. arXiv preprint arXiv:2509.13186 (2025)
24. Oest, A., Safei, Y., Doupé, A., Ahn, G.J., Wardman, B., Warner, G.: Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis. In: 2018 APWG Symposium on Electronic Crime Research (eCrime). pp. 1–12. IEEE (2018). https://doi.org/10.1109/ECRIME.2018.8376206
25. Saka, T., Vaniea, K., Kökciyan, N.: Context-based clustering to mitigate phishing attacks. In: Proceedings of the 15th ACM Workshop on Artificial Intelligence and Security. pp. 115–126 (2022). https://doi.org/10.1145/3560830.3563728
26. Sanchez-Rola, I., Bilge, L., Balzarotti, D., Buescher, A., Efstathopoulos, P.: Rods with laser beams: Understanding browser fingerprinting on phishing pages. In: 32nd USENIX Security Symposium (USENIX Security 23). pp. 4157–4173. USENIX Association, Anaheim, CA (Aug 2023), https://www.usenix.org/conference/usenixsecurity23/presentation/sanchez-rola
27. Santos, J.M., Embrechts, M.: On the use of the adjusted rand index as a metric for evaluating supervised classification. In: International conference on artificial neural networks. pp. 175–184. Springer (2009). https://doi.org/10.1007/978-3-642-04277-5_18

28. Thonnard, O., Bilge, L., O'Gorman, G., Kiernan, S., Lee, M.: Industrial espionage and targeted attacks: Understanding the characteristics of an escalating threat. In: International workshop on recent advances in intrusion detection. pp. 64–85. Springer (2012). https://doi.org/10.1007/978-3-642-33338-5_4
29. Thonnard, O., Dacier, M.: A strategic analysis of spam botnets operations. In: Proceedings of the 8th Annual Collaboration, Electronic messaging, Anti-Abuse and Spam Conference. pp. 162–171 (2011). https://doi.org/10.1145/2030376.2030395
30. Yager, R.R.: On ordered weighted averaging aggregation operators in multicriteria decisionmaking. IEEE Transactions on systems, Man, and Cybernetics **18**(1), 183–190 (2002). https://doi.org/10.1109/21.87068
31. Yearwood, J., Mammadov, M., Banerjee, A.: Profiling phishing emails based on hyperlink information. In: 2010 International Conference on Advances in Social Networks Analysis and Mining. pp. 120–127. IEEE (2010). https://doi.org/10.1109/ASONAM.2010.56
32. Zawoad, S., Dutta, A.K., Sprague, A., Hasan, R., Britt, J., Warner, G.: Phish-net: investigating phish clusters using drop email addresses. In: 2013 APWG eCrime Researchers Summit. pp. 1–13. IEEE (2013). https://doi.org/10.1109/eCRS.2013.6805777

Appendix A

For fuzzy image hashes, we use approximate rather than exact intersection. Given two sets of fuzzy hashes A and B:

- Intersection ($|A \cap B|$) is the number of unique pairs (a, b) with normalized Hamming distance < 0.1. Each $b \in B$ can be matched at most once (greedy one-to-one matching).
- Union is computed as: $|A \cup B| = |A| + |B| - |A \cap B|$

This intersection is order-dependent and may underestimate the maximum match count, but it is sufficient here as a robust visual-similarity proxy.

Appendix B

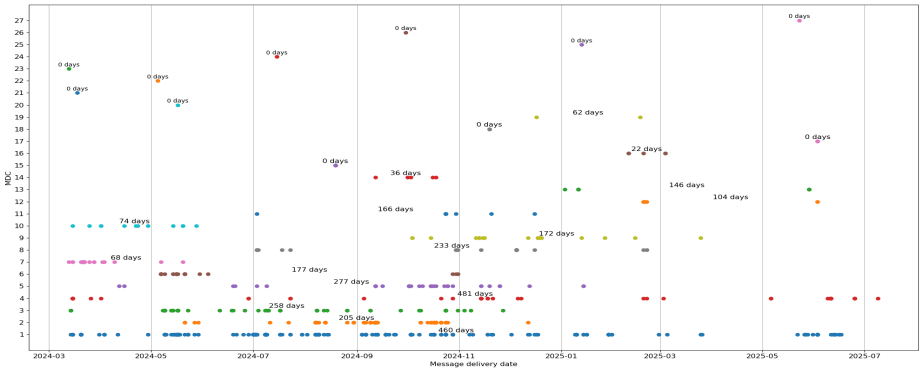


Fig. 7: Temporal distribution of the obtained MDCs.