

Article

Leveraging Transformer Embeddings for Real-Time Discovery of Malicious URL Campaigns in the Generative-AI Threat Era

Georgios Feretzakis , Dimitrios Karapiperis  and Sarandis Mitropoulos 

School of Science and Technology, Hellenic Open University, 263 31 Patras, Greece; dkarapiperis@eap.gr (D.K.); smitropoulos@eap.gr (S.M.)

* Correspondence: georgios.feretzakis@ac.eap.gr

Abstract

The proliferation of algorithmically generated malicious URLs presents a critical challenge for modern cybersecurity, requiring a shift from syntactic pattern-matching toward semantic understanding grounded in pre-trained transformer foundations. Building on our prior work on billion-scale semantic search and density-based campaign clustering, this paper presents a unified, deployable framework for real-time campaign discovery. The framework converts raw URL streams into dense Sentence-BERT embeddings and couples approximate nearest neighbor search with online density-based clustering, discovering emerging campaigns without prior knowledge of their number or shape. Our central finding is that the choice of semantic representation is decisive: a domain-focused embedding strategy yields near-perfect campaign separation, substantially outperforming full-URL representations. On live, in-the-wild threat feeds, the domain-focused representation recovers all 944 discovered campaigns at an Adjusted Rand Index of 0.990 and a mean campaign recall of 1.000, at 0.10 ms per URL. Under identical clustering, the full-URL representation reaches an Adjusted Rand Index of only 0.510 and recovers fewer than half of the campaigns. We add a SHAP-based explainability layer that attributes discovery decisions to interpretable structural patterns, and we expose the whole pipeline as an operational system with a REST interface for single- and batch-URL analysis. The encoder we employ is a discriminative representation model rather than a generative one; what we contribute is the representation-and-modeling discipline this setting demands, and the resulting guidance transfers to threat intelligence in an era of generatively produced attack campaigns.

Keywords: phishing detection; transformer embeddings; Sentence-BERT; semantic search; approximate nearest neighbor; density-based clustering; campaign discovery; explainable AI; real-time threat intelligence; generative AI



Academic Editor: Tania Yankova

Received: 29 June 2026

Revised: 21 July 2026

Accepted: 24 July 2026

Published: 27 July 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

Malicious Uniform Resource Locators (URLs) remain one of the most pervasive vectors in contemporary cyberattacks, underpinning phishing, malware distribution, and command-and-control (C2) communication. For much of the past two decades, the dominant line of defense has been syntactic: curated blocklists of known-malicious domains together with hand-crafted regular-expression rules over the URL string. While effective against static, previously observed threats, this paradigm has become increasingly brittle. Modern adversaries no longer rely on a small, stable set of indicators; instead, they orchestrate large campaigns comprising thousands or even millions of short-lived URLs. A central

enabler of this strategy is the Domain Generation Algorithm (DGA), which produces a continuous stream of pseudo-random domains that serve as rendezvous points with C2 infrastructure and render manual blocklist curation futile [1]. Adversaries further obscure their URLs through typosquatting, homoglyph substitution, and the use of compromised but otherwise legitimate hosts, defeating filters that reason only over surface characters [2].

The sheer volume, ephemerality, and polymorphism of these threats imply that robust detection must reason about the *intent* of a URL rather than its literal string. Pre-trained transformer language models offer precisely this capability. By projecting semi-structured URL strings into dense, high-dimensional embeddings, such models induce a representation space in which geometric proximity reflects semantic similarity: two syntactically different variants of the same phishing attempt are mapped to nearby vectors [3,4]. This representation enables two complementary operations at scale. First, approximate nearest neighbor (ANN) search supports rapid *retrieval* of semantically related URLs from massive historical corpora. Second, density-based clustering supports *discovery* of coherent groups of related attacks.

Retrieval and discovery are not interchangeable. ANN retrieval is fundamentally reactive; it answers the query “find more URLs like this one” but presupposes that an analyst already holds a malicious lead. The pressing operational need is for proactive discovery—continuously surveying the global, high-velocity stream of observed URLs to answer “which campaigns are emerging, evolving, and fading right now?” Achieving this in production requires more than a clustering algorithm in isolation; it requires an end-to-end system that embeds URLs, indexes them for scalable similarity search, maintains an evolving summary of the stream, surfaces interpretable explanations for its decisions, and exposes all of this through an operational interface.

Relation to our prior work and contributions.

This paper builds on, and is careful to delineate itself from, three of our previously published systems. In PhishGraph [5], we introduced a disk-aware ANN index, built on a Vamana proximity graph, for a billion-scale semantic URL search. In PhishCluster [6], we introduced an online–offline, density-based algorithm that discovers campaigns from a stream of URL embeddings, evaluated on a synthetic billion-point dataset designed to mimic campaign dynamics. In a complementary conference contribution [7], we studied high-throughput generation of semantic embeddings for real-time threat campaigns. The present work does not represent the ANN index, the clustering algorithm, or the embedding-throughput pipeline as novel. Rather, its contributions are as follows:

- A *unified, end-to-end, deployable* framework that composes semantic embedding, large-scale ANN search, online density-based discovery, and explainability into a single operational system, with a REST interface for real-time single- and batch-URL analysis.
- The identification and empirical validation, *on live real-world threat feeds*, of *domain-focused semantic representation* as a decisive modeling principle for campaign discovery. It substantially outperforms full-URL representations on standard clustering-quality metrics.
- An integrated SHAP-based explainability layer that attributes discovery to interpretable structural patterns and supports analyst trust.
- A characterization of the framework as a working system validated against current, in-the-wild data rather than solely against in-distribution benchmarks.

Collectively, these reframe the problem as one of *representation and modeling* for threat intelligence in the era of generative AI—the theme of this Special Issue. We use the term *generative AI* to describe the adversarial setting rather than our method: modern campaigns are themselves algorithmically—and, increasingly, generatively—produced, from DGA domains to LLM-authored lures, and tracking them at scale is fundamentally

a representation-and-modeling problem. The Sentence-BERT/MiniLM encoder we employ is a discriminative representation model, not a generative one; what transfers to the generative-AI setting is this modeling discipline, not a generative component.

The remainder of this paper is organized as follows: Section 2 reviews related work. Section 3 details the framework's representation, search, clustering, explainability, and system components, together with the datasets and evaluation protocol. Section 4 reports the experimental evaluation. Section 5 discusses implications and limitations, and Section 6 concludes.

2. Related Work

2.1. Lexical and Feature-Based URL Detection

Early and still widely deployed approaches to malicious-URL detection extract hand-engineered lexical and host-based features—URL length, token counts, character entropy, presence of suspicious top-level domains, IP-literal hosts, and similar indicators—and feed them to supervised classifiers such as decision trees, random forests, and support vector machines [2]. These methods are fast and interpretable, and remain effective against known patterns. However, they are inherently reactive: feature templates encode the obfuscation tricks observed at design time, and adversaries can evade them by introducing previously unseen surface variations. Their lexical nature also limits their ability to recognize that two superficially different URLs pursue the same malicious intent. Recent reviews of machine-learning URL detection reach the same verdict, singling out dataset freshness, adversarial drift, and the brittleness of hand-crafted features as the principal open challenges for feature-based defenses [2].

2.2. Semantic Representations for Security

To move beyond surface features, recent work represents URLs and other semi-structured security artifacts using dense embeddings produced by pre-trained transformer models. Sentence-BERT [3] adapts BERT with a Siamese architecture to yield sentence-level embeddings whose cosine geometry reflects semantic similarity, and distilled variants such as MiniLM [4] retain much of this quality at a fraction of the inference cost—an important property for high-velocity security streams. Embedding URLs in such a space allows semantically related threats to be grouped regardless of syntactic divergence, providing the substrate on which both retrieval and clustering operate. A parallel line of work adapts transformers to URLs directly: URLTran fine-tunes BERT/Roberta for phishing-URL classification [8], large-scale URL-specific pre-training further improves detection at low false-positive rates [9], and domain- and URL-aware encoders such as DomURLs_BERT specialize the tokenizer and pre-training objective to this text type [10]. Such models can outperform a general-purpose encoder, at the cost of task-specific training and maintenance. Our study is deliberately complementary. Rather than train a URL-specific model, we ask what an *off-the-shelf* sentence encoder already achieves when the *input* is chosen well, and we find that encoding the registered domain alone suffices for near-perfect campaign separation—a result orthogonal to, and combinable with, URL-specific pre-training.

2.3. Large-Scale Approximate Nearest-Neighbor Search

Operating over embeddings at web scale requires efficient similarity search. Graph-based indices such as HNSW [11] offer excellent in-memory query performance, while DiskANN [12] demonstrated that a single SSD-resident Vamana graph can serve billion-point datasets at low latency, and systems such as SPFresh [13] extended this to incremental, in-place updates for billion-scale streaming search. Complementary lines of work reduce memory footprint through product and anisotropic quantization [14,15] and explore

attribute-constrained search [16]. Our PhishGraph index [5] adapts these foundations to the cybersecurity setting, integrating structured metadata into the graph topology and supporting streaming, adversarially skewed workloads; in the present framework, it provides the retrieval substrate.

2.4. Density-Based and Streaming Clustering

Discovering campaigns—as opposed to retrieving neighbors—calls for clustering that does not assume a fixed number of spherical groups. DBSCAN [17] recovers arbitrarily shaped clusters as density-connected regions separated by low-density noise, and HDBSCAN [18] generalizes this to varying densities through a hierarchical formulation. For unbounded streams, online density-based methods such as DenStream [19] and systems like DBStream [20] summarize data into evolving micro-clusters that are periodically consolidated. Our PhishCluster algorithm [6] combines an ANN-accelerated online maintenance phase with a hierarchical density-based offline phase to discover campaigns from URL embedding streams; the present framework adopts it as its discovery component.

2.5. Explainable AI in Cybersecurity

Operational adoption of machine-learning detectors depends on the interpretability of their decisions. Model-agnostic methods such as LIME [21] and SHAP [22] attribute a prediction to the contributions of individual input features, enabling analysts to understand and audit automated alerts. Explainability has since become a substantial theme in security; surveys catalog its use across intrusion detection, malware analysis, and phishing [23], and recent studies apply SHAP-based attribution specifically to web-phishing classification to expose which URL and host features drive a decision [24]. A recurring lesson is that feature attributions are most trustworthy when computed over a compact, semantically meaningful feature set rather than an opaque high-dimensional space. Within this project, we previously applied LIME and SHAP to URL threat detection in a healthcare setting [25]; here we integrate SHAP-based attribution directly into the discovery framework, over an interpretable set of structural URL attributes, linking campaign assignments to human-readable patterns.

2.6. Positioning

Recent systems attack neighboring problems with heavier machinery—for example, reference- and knowledge-graph-based phishing detection that couples large language models with brand knowledge [26]—but these target per-URL classification against known brands rather than unsupervised discovery of *emerging* campaigns from a live stream. While each of the components we build on is individually mature, their composition into a single, explainable, deployable system for proactive campaign discovery—and, in particular, a principled study of how the choice of semantic representation governs discovery quality on live data—has received comparatively little attention. The present work addresses this gap.

3. Materials and Methods

3.1. Framework Overview

The proposed framework is an end-to-end pipeline that transforms a high-velocity stream of raw URLs into an interpretable, continuously updated view of active campaigns. Each incoming URL is first encoded into a dense semantic embedding (Section 3.2). Embeddings are inserted into a disk-aware ANN index that supports both low-latency similarity queries and streaming updates (Section 3.3). In parallel, an online density-based procedure summarizes the embedding stream into evolving micro-clusters, which an on-demand offline phase consolidates into campaigns (Section 3.4). For each discovered campaign, an

explainability module attributes the grouping to interpretable URL features (Section 3.5). The entire pipeline is exposed as an operational service through a REST interface supporting real-time single- and batch-URL analysis (Section 3.6). Section 3.7 describes the datasets and evaluation protocol.

3.2. Semantic Representation of URLs

We encode URLs using a distilled Sentence-BERT model, all-MiniLM-L6-v2 [3,4], which maps an input string to a 384-dimensional vector; embeddings are L_2 -normalized so that Euclidean distance is monotonically related to cosine similarity. A central design decision concerns *what* text is embedded. We contrast two representations:

- **Full-URL representation:** the complete URL string—scheme, host, path, query, and fragment—is embedded directly.
- **Domain-focused representation:** only the registered domain (the host component, obtained by standard URL parsing) is embedded, discarding path, query, and fragment.

Our working hypothesis is that, for campaign discovery, the domain-focused representation is the more informative unit. Members of a coordinated campaign typically share infrastructure at the domain level—variants generated by a common kit or DGA seed—while their paths and query strings vary widely and inject high-variance, campaign irrelevant tokens that dilute the semantic signal. Section 4 tests this hypothesis quantitatively. We stress that this is a comparison of *inputs*, not of *encoders*; both representations are produced by the *same* pre-trained model under identical settings and differ only in what text is fed to it. The question is therefore not whether the encoder is optimized for URLs, but which unit of the URL carries the campaign invariant signal.

3.3. Approximate Nearest-Neighbor Search

Scalable retrieval over the embedding space is provided by PhishGraph [5], a disk-aware ANN index built on a Vamana proximity graph optimized for SSD residency. We summarize it here only to the extent needed for the pipeline to be self-contained; full details appear in the original work. PhishGraph augments the base graph with a fusion distance that integrates structured metadata into the topology for constrained search, a dual-mode update mechanism that combines high-throughput batch consolidation with low-latency in-place insertions for streaming data, and an adaptive maintenance policy that mitigates performance degradation under skewed, adversarial query workloads. In the present framework, PhishGraph serves both the analyst-facing retrieval function and the neighbor queries used to accelerate online clustering.

3.4. Online Density-Based Campaign Discovery

Campaign discovery is performed by PhishCluster [6], which adopts a two-phase, online–offline architecture. In the online phase, an ANN-accelerated maintenance algorithm processes the embedding stream and summarizes it into compact, evolving Campaign Micro-Clusters (CMCs), each capturing a dense, temporally local group of related embeddings. In the on-demand offline phase, a hierarchical density-based procedure is applied to the CMC summaries, recovering arbitrarily shaped, varying density campaigns without requiring their number to be specified in advance and labeling sparse regions as noise. Following the formalization in the original work, two URLs are treated as related when their embeddings are both spatially proximate and temporally co-located, and a campaign is a density-connected set of such related attacks.

3.5. Explainability Layer

To support analyst trust and auditability, the framework integrates a SHAP-based explainability module [22]. Operating over an interpretable set of structural URL attributes—such as host and path lengths, subdomain count, character entropy, presence of suspicious top-level domains, and related indicators—the module attributes each classification and campaign assignment decision to the contributions of individual features. This yields both per-decision explanations and an aggregate view of which structural patterns drive the system’s behavior, complementing the otherwise opaque embedding space. The choice of SHAP follows our earlier explainability work in the project [25].

3.6. System Architecture and Deployment

The framework is realized not as an offline experiment but as an operational application. A backend service exposes a REST API with endpoints for real-time single-URL analysis, batch analysis, health monitoring, and aggregate statistics; a request returns the predicted class, a calibrated risk score, the extracted feature set, and the associated explanation. A web dashboard provides interactive single- and batch-URL analysis, model-training controls, and historical and statistical views for security analysts. This deployment provides a concrete, running artifact against which per-URL latency and end-to-end throughput are measured under realistic conditions. We characterize it as a deployable research prototype rather than a hardened production service: it validates computational efficiency and end-to-end operation, while engineering concerns such as horizontal scalability, fault tolerance, and sustained high-rate streaming insertion are deployment questions we do not evaluate here (Section 5).

3.7. Datasets and Evaluation Protocol

We evaluate the framework on current, in-the-wild data drawn from publicly available threat intelligence feeds, in order to assess generalization beyond in-distribution benchmarks. Malicious URLs are sourced from live feeds—OpenPhish and URLhaus—while benign traffic is drawn from the Tranco research-oriented top-sites ranking, which is designed to provide a stable, reproducible baseline of legitimate domains [27–29]. Because in-the-wild feeds do not ship with campaign labels, ground-truth campaigns are constructed by a single reproducible rule: malicious URLs are grouped by their registered domain (eTLD + 1), and a group is retained as a campaign only if it contains at least three URLs spanning at least two distinct paths. The path requirement is deliberate: it guarantees that every campaign genuinely exercises path variation, so that the comparison between full-URL and domain-focused representations is non-trivial rather than decided purely by construction (Section 5 returns to this point). The released implementation additionally provides a typosquat/homoglyph stress mode in which the registered domain varies *within* a campaign, for evaluating semantic rather than string-level matching.

We compare the domain-focused and full-URL representations under identical clustering, and against representative baselines—DBSCAN and HDBSCAN over full-URL embeddings, and K-Means over lexical URL features. Clustering quality is measured by the Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) against the constructed ground truth; campaign detection fidelity by recall; and efficiency by per-URL processing latency and end-to-end throughput. All experiments are run on the deployed system described in Section 3.6, so that reported efficiency reflects realistic deployment conditions.

4. Results

All results below are from a single live snapshot of the threat feeds collected on 29 June 2026; because the feeds update continuously, exact counts vary slightly between collections. The snapshot comprises 28,932 URLs: 8932 malicious URLs organized into 944 ground-truth campaigns (OpenPhish and URLhaus, retained after the ≥ 3 -URL, ≥ 2 -distinct-path filter of Section 3.7) together with 20,000 Tranco benign domains as background traffic. The controlled comparison applies DBSCAN—the density-based clusterer at the core of the offline discovery phase—directly to each of the two embedding variants, sweeping its neighborhood radius ϵ , with HDBSCAN over the same embeddings and K-Means over lexical URL features as baselines; the density parameter $\text{min_samples} = 3$ throughout. This value—the standard minimal setting for a density *core* (a point together with two neighbors)—should not be conflated with the ≥ 3 -URL threshold used to *define* a ground-truth campaign in Section 3.7. The two are independent knobs: min_samples governs cluster formation, whereas the labeling threshold governs which domains are statistically substantial enough to count as campaigns. We hold min_samples fixed at this minimal value across all methods so that the comparison isolates the effect of the *representation* rather than of density tuning. Clustering quality is reported as Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI) against the constructed ground truth; campaign detection fidelity by two complementary measures—campaign recall, the mean fraction of each ground-truth campaign captured by its single dominant predicted cluster, and the number of campaigns recovered, those with at least half of their members in one non-noise cluster (the Campaigns column of Table 1)—and efficiency as per-URL processing latency on the deployed system.

Table 1. Campaign-discovery quality and per-URL latency on the live threat-feed snapshot of 29 June 2026. Each configuration is shown at its own optimal DBSCAN ϵ ; HDBSCAN and K-Means require none. Best value per column in bold.

Representation (Method)	ϵ	ARI	NMI	Recall	Campaigns	ms/URL
Domain-focused (DBSCAN)	0.2	0.990	0.993	1.000	944/944	0.10
Domain-focused (HDBSCAN)	—	0.277	0.627	0.988	944/944	6.35
Full-URL (DBSCAN)	0.5	0.510	0.494	0.404	457/944	0.11
Full-URL (HDBSCAN)	—	0.107	0.424	0.544	537/944	7.57
Lexical (K-Means)	—	0.009	0.464	0.490	455/944	0.24

4.1. Representation Is Decisive: Domain-Focused vs. Full-URL

Table 1 reports each configuration at its own optimal ϵ (selected by the sweep in Section 4.2), so that every method is shown under its most favorable setting rather than a setting tuned for one representation. The domain-focused representation recovers *all* 944 campaigns—each with a dominant cluster, at mean campaign recall = 1.000—with ARI = 0.990 and NMI = 0.993, whereas the identical clustering applied to full-URL embeddings reaches only ARI = 0.510, recovers 457 of the 944 campaigns, and attains a mean campaign recall of just 0.404. Lexical K-Means is effectively random for this task (ARI = 0.009). The gap is not marginal: changing only *what* is embedded—the registered domain versus the full URL string—raises campaign recall by a factor of roughly 2.5 and ARI by nearly 0.5 in absolute terms, under otherwise identical conditions. For calibration, consider the trivial baseline of grouping URLs by an exact hash of their registered domain. On this benchmark that baseline is perfect *by construction*—the ground truth *is* the registered domain—which is precisely why exact-match accuracy on the standard feed is not, on its own, evidence that semantic embedding helps. The informative question is what happens when the domain string is *not* stable within a campaign. There, exact hashing degenerates to one cluster per distinct string, whereas a semantic encoder can still map typo- and

homoglyph-variants of a domain to nearby points; the released stress mode (Section 3.7) is built to exercise exactly this regime. We therefore read the *full-URL* comparison, not the exact-match baseline, as the substantive test on the standard feed: full-URL embeddings *contain* the domain yet *fail*, because path and query tokens dominate the representation.

Figure 1 plots these scores together with the corresponding per-URL latencies, and Figures 2 and 3 explain the mechanism behind them. Figure 2 is the most direct: under the domain-focused representation, within-campaign pairwise distances concentrate sharply near zero—campaign members share a registered domain and therefore map to nearly identical embeddings—while between-campaign distances occupy a well-separated band beyond 0.6, leaving a wide margin in which essentially any density threshold isolates campaigns cleanly. Under the full-URL representation the within-campaign distribution spreads across the whole $[0.1, 1.0]$ range and overlaps the between-campaign mass, so no single radius separates the two: exactly the failure reflected in the full-URL scores. The same contrast is visible geometrically in the UMAP projection (Figure 3): domain-focused embeddings collapse each campaign to a compact, well-separated locus while benign traffic forms a single dense background mass, whereas full-URL embeddings smear campaigns into overlapping filaments as high-variance path and query tokens pull members of one campaign apart and intermix them with members of others.

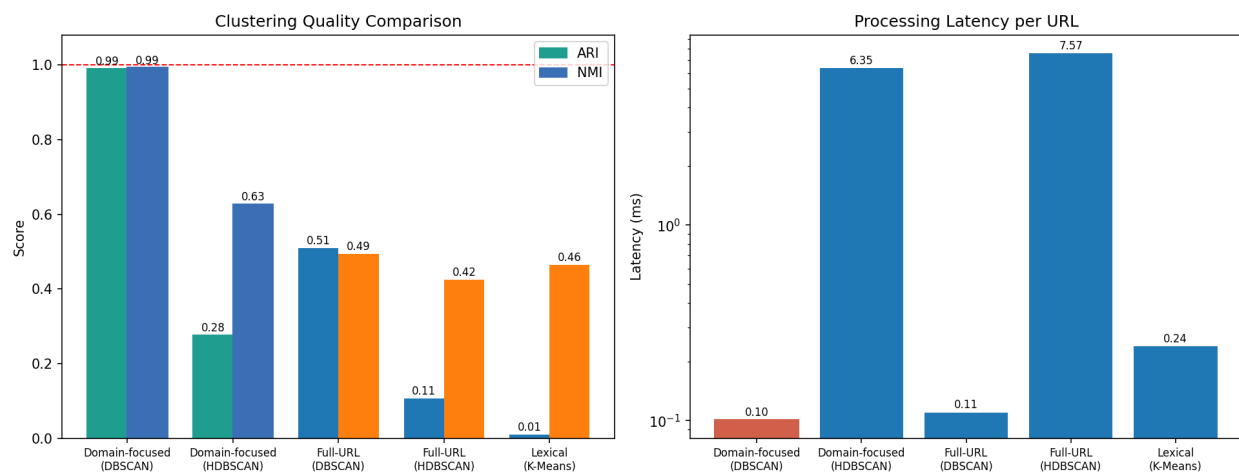


Figure 1. Clustering quality (left) and per-URL latency (right, log scale) for the five configurations of Table 1.

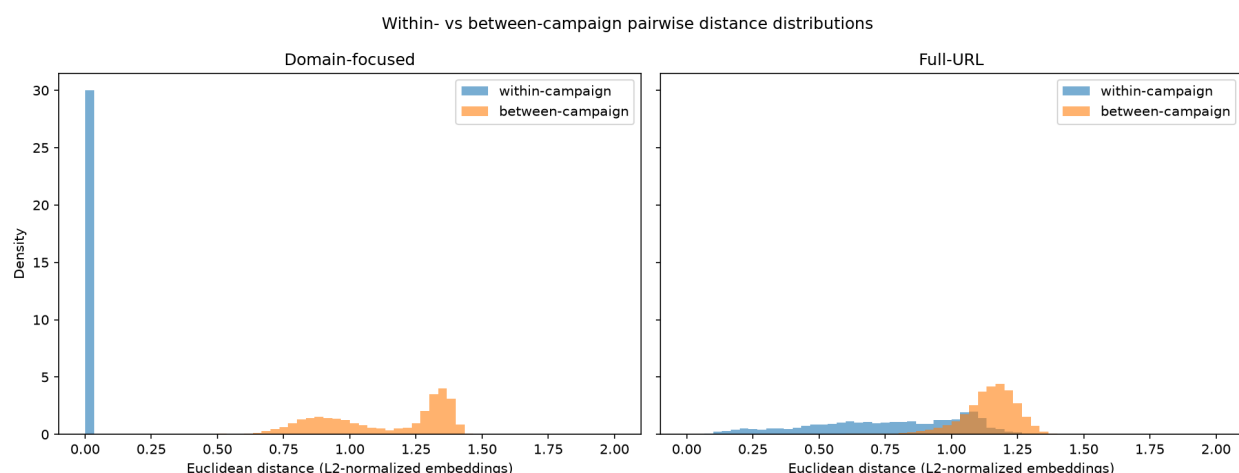


Figure 2. Within- versus between-campaign pairwise Euclidean distances (L_2 -normalized) for the domain-focused (left) and full-URL (right) representations.

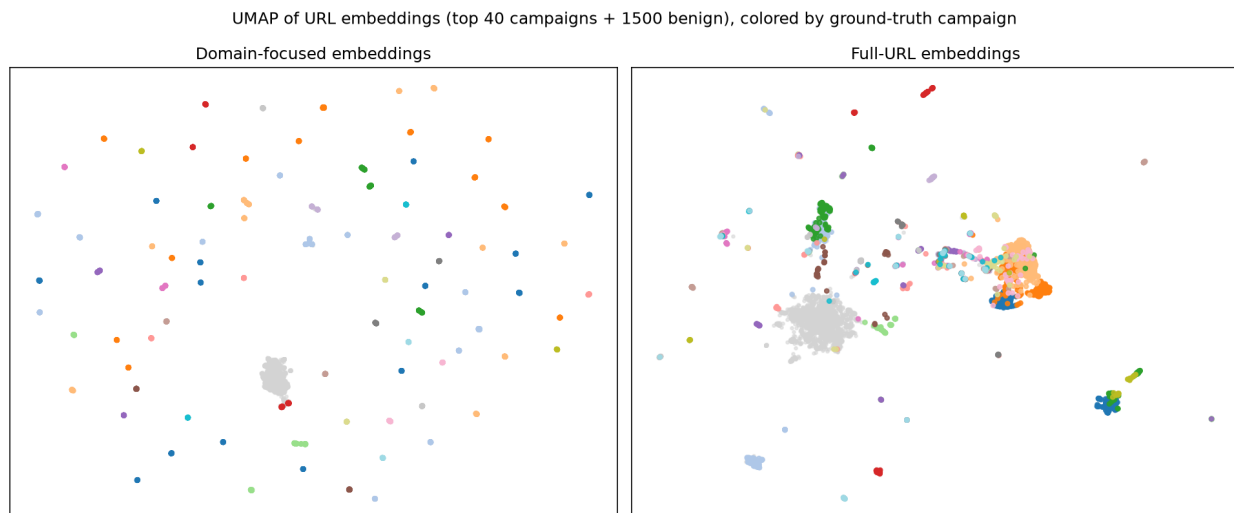


Figure 3. UMAP projection of the embedding space (40 largest campaigns plus a benign subsample, colored by ground-truth campaign, where each distinct color denotes a separate campaign and the grey points are benign background traffic) for the domain-focused (**left**) and full-URL (**right**) representations.

The HDBSCAN rows merit comment. Although HDBSCAN requires no ϵ and is often preferable for varying density data, here it attains high recall (0.988 over domain-focused embeddings) but low ARI (0.277), fragmenting the 944 campaigns into 2827 micro-clusters. The cause is the same property that *helps* DBSCAN—campaign members collapse to near-identical points—which interacts poorly with HDBSCAN’s mutual-reachability construction over large sets of duplicate points, splitting single campaigns across many leaf clusters. We therefore treat HDBSCAN as an ϵ -free baseline and adopt ϵ -swept DBSCAN as the operating method.

4.2. Robustness to the Clustering Threshold

A natural objection is that the advantage might hinge on a fortunate choice of ϵ . Figure 4 shows the opposite. For the domain-focused representation, campaign recall is exactly 1.000 across the entire swept range $\epsilon \in [0.2, 0.8]$, while ARI stays high for small ϵ (0.990, 0.971, and 0.949 at $\epsilon = 0.2, 0.3, 0.4$) before declining gracefully as larger radii begin to merge distinct campaigns. The full-URL representation instead exhibits an irreducible tension: its ARI peaks at only 0.510 ($\epsilon = 0.5$), where recall is just 0.404; increasing ϵ to 0.8 lifts recall to 0.938 but collapses ARI to -0.107 as clusters merge indiscriminately. There is no radius at which the full-URL representation achieves both high recall and high purity, whereas the domain-focused representation achieves both across a broad, easily chosen range. The representation advantage is therefore structural, and not an artifact of tuning.

Robustness across encoders. The advantage could in principle be specific to the distilled MiniLM encoder. To test this, we repeated the comparison on a fresh live snapshot (749 ground-truth campaigns) using a second, substantially larger off-the-shelf encoder, all-mpnet-base-v2 [30] (roughly 110M parameters versus 22M for MiniLM), under the identical protocol and ϵ -sweep. Table 2 shows that the representation gap is essentially encoder-invariant: both encoders reach a domain-focused ARI of about 0.96 and a full-URL ARI of only 0.54–0.59 at their best ϵ . The larger encoder yields at most a marginal improvement on the full-URL representation (ARI 0.541 $\rightarrow$ 0.591) and none on the already near-perfect domain-focused representation, and it does not narrow the gap. The bottleneck is therefore *what* is embedded, not the capacity of the encoder that embeds it. Absolute values differ slightly from Table 1 because the feeds update continuously, as noted above.

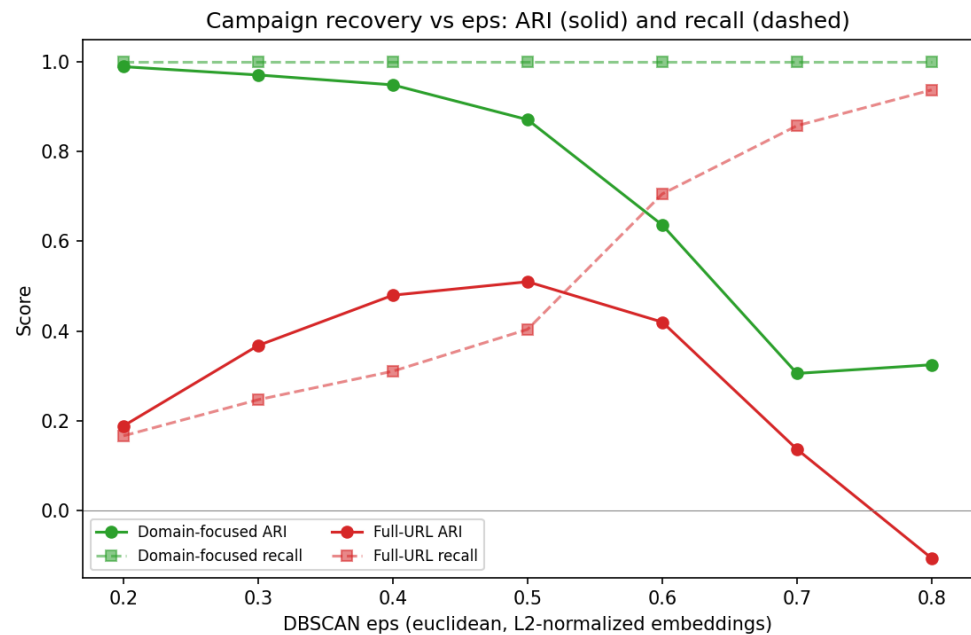


Figure 4. Campaign-recovery quality versus DBSCAN radius ϵ : ARI (solid) and recall (dashed) for each representation.

Table 2. Robustness of the representation effect across encoders: best- ϵ campaign discovery quality for the domain-focused and full-URL inputs under two off-the-shelf Sentence-BERT encoders, on a fresh live snapshot (749 campaigns).

Representation	Metric	MiniLM-L6	MPNet-Base
Domain-focused	ARI	0.960	0.961
	Recall	0.905	0.907
Full-URL	ARI	0.541	0.591
	Recall	0.446	0.395

4.3. Real-Time Performance

Efficiency follows the same ordering (Table 1, last column). On the deployed system, domain-focused DBSCAN clusters at 0.10 ms per URL—on the order of 10^4 URLs per second on a single thread—because operating on registered domains both shortens the encoded string and collapses many URLs onto shared points, shrinking the effective problem size. Full-URL DBSCAN is comparable per URL (0.11 ms) but fails on quality as shown above, while the HDBSCAN baselines are one to two orders of magnitude slower (6.3–7.6 ms per URL) with no accuracy benefit on this task. Crucially, the most accurate configuration is also the fastest (Figure 1): the pipeline does not force a speed–accuracy trade-off, which is what makes it suitable for real-time operation on streaming feeds.

4.4. Explainability of Decisions

Beyond discovery, the deployed system attaches a SHAP-based explanation to every decision over the interpretable structural feature set of Section 3.5, yielding both per-decision attributions and an aggregate view of which attributes drive behavior. Consistent with the representation finding, host- and domain-level structural attributes dominate the aggregate attribution, while path- and query-derived features contribute comparatively little—an independent, model-level corroboration that the campaign discriminative signal resides at the domain rather than the path level. This explainability layer follows the project’s earlier work on interpretable phishing detection [25]. Table 3 makes this quantita-

tive. Aggregated over all scored URLs, domain structure features contribute the largest single share of total absolute SHAP attribution (66.8%), and host-tied security indicators such as the top-level domain and HTTPS usage account for a further 19.8%, while URL-structure, character-pattern, and keyword features together contribute only 13.3%. This is an independent, model-level echo of the representation result: the campaign discriminative signal resides predominantly at the domain and host level.

Table 3. Aggregate SHAP feature-category contributions to the detection decision.

Feature Category	Share of Total SHAP
Domain structure	66.8%
Security indicators (TLD, HTTPS)	19.8%
URL structure	7.4%
Character patterns	3.2%
Keywords	2.7%

4.5. Operational Validation

The same deployment additionally runs a supervised single-URL classifier behind the REST API. We mention it only as end-to-end deployment evidence—confirmation that the pipeline serves live predictions—and it is distinct from, and not evaluated as part of, the unsupervised campaign discovery contribution that is the subject of this paper.

5. Discussion

The central empirical result is that, for real-time phishing campaign discovery, the choice of what to embed dominates the choice of clustering algorithm. Embedding the registered domain rather than the full URL turns a poorly separated problem into an almost trivially separable one: domain-focused density clustering recovers every campaign at ARI 0.990, while the same clustering over full-URL embeddings recovers fewer than half. The mechanism is visible in the distance distributions of Figure 2, where members of a coordinated campaign share infrastructure at the domain level, so their domain embeddings nearly coincide, whereas their paths and query strings vary widely and, when embedded, inject high-variance, campaign irrelevant tokens that wash out the shared signal. This generalizes into a simple design principle for embedding-based security analytics—encode the campaign invariant unit, not the most information-rich string—which we expect to transfer to other settings where coordinated activity shares a stable substrate beneath high-variance surface form.

It is important to be precise about what this experiment does and does not show, because the ground-truth campaigns are themselves defined by registered domain. It would be circular to read the result as evidence that “campaigns share domains”—that is assumed, not discovered. The non-trivial finding is the opposite direction: full-URL embeddings *contain* the domain and yet *fail*, because path and query tokens dominate the representation. The ≥ 2 -distinct-path requirement on every campaign (Section 3.7) ensures the contrast is substantive rather than mechanical: each campaign genuinely exhibits path variation that the full-URL representation must contend with, and it is precisely this variation that fragments it. The released typosquat/homoglyph mode pushes further: it constructs campaigns whose registered domains vary *within* the campaign—a form of multi-domain campaign—so that domain-focused matching must generalize semantically rather than match a string. This is also the natural adversarial-evasion setting; attackers routinely mutate domains through typosquatting and homoglyph substitution to defeat exact-match filters, and such perturbations are known to degrade even transformer-based URL classifiers unless the model is explicitly hardened [8]. We provide this mode as a

reproducible stress test and adversarial probe, and report a systematic evaluation over it—together with compromised-host and redirect-chain cases—as future work rather than as a headline result here.

These findings position the present work relative to its components. PhishGraph [5] contributes billion-scale approximate nearest neighbor search and PhishCluster [6] contributes streaming density-based discovery; each was previously evaluated in isolation and, in the case of PhishCluster, on synthetic billion-point workloads. The contribution here is twofold: a unified, explainable, and *deployed* composition of these components into a single operational pipeline, and the representation finding established on *live, in-the-wild* feeds rather than synthetic data. This also distinguishes the present work from our recent healthcare-focused study [31], which specialized a related domain-focused pipeline to a sector-specific benchmark of healthcare-targeted campaigns; consistent with that study, host- and domain structure features again dominate the interpretability proxy here, while the evidence in the present paper concerns general, live threat-feed traffic and foregrounds the modeling question of representation rather than a single sector.

Several threats to validity remain, and we state them plainly. *Ground truth.* Labels derive from a domain-based rule; although the ≥ 2 -distinct-path requirement makes the representation contrast non-trivial, a fully label-independent ground truth—analyst-verified campaign membership, phishing-kit fingerprints, or CERT advisories—would further strengthen the claim, and integrating such sources is planned. *Temporal scope.* Our evaluation uses a single live snapshot; real campaigns emerge, mutate, and fade over days to months, so a longitudinal study across many snapshots is needed to characterize stability over time. *Metric sensitivity.* ARI is influenced by the large benign mass correctly assigned to noise, which can flatter absolute values; we therefore foreground recall and the threshold-robustness analysis of Figure 4, neither of which shares this sensitivity. *Encoder coverage.* The main results use a single distilled encoder (all-MiniLM-L6-v2); we verified that the representation effect persists under a substantially larger encoder (all-mpnet-base-v2; Section 4.2, Table 2), and confirming it on URL-adapted encoders (e.g., URL-specific transformers [8,10]) remains as future work. *Baseline coverage.* We compare against DBSCAN, HDBSCAN, and lexical K-Means; stronger discovery baselines—spectral, deep, and graph-based clustering, and recent phishing-campaign systems—would sharpen the comparison. *Campaign complexity.* Campaigns are defined at the registered-domain level; harder real-world structures such as compromised legitimate hosts, multi-domain campaigns, and redirect chains are only partially represented (the stress mode captures within-campaign domain variation) and warrant dedicated evaluation. *Deployment surface.* Evidence here concerns public threat intelligence feeds; generalization to other vantage points such as email gateways or enterprise traffic remains to be shown. *Data noise.* Public feeds may contain mislabeled or duplicated entries, and Tranco domains are assumed benign. *HDBSCAN.* Its under-performance is specific to the near-duplicate geometry induced by domain embeddings and should not be read as a general claim about HDBSCAN.

6. Conclusions

We presented a unified, explainable, and deployable framework for real-time discovery of malicious URL campaigns that composes transformer embeddings, approximate nearest neighbor search, online density-based clustering, and SHAP-based explanation into a single operational system. On a live threat-feed snapshot, the framework’s domain-focused representation recovered all 944 ground-truth campaigns at ARI 0.990 while clustering at 0.10 ms per URL, and did so robustly across a wide range of the clustering threshold—whereas the conventional full-URL representation could not achieve high recall and high purity at any threshold. The practical lesson is concise: for embedding-based campaign

discovery, encode the campaign invariant unit. Future work includes comparing further transformer foundation models beyond the MiniLM and MPNet encoders evaluated here (in particular URL-adapted encoders), benchmarking against stronger discovery base-lines, validating against analyst-verified ground truth and over longer, multi-snapshot deployment windows, and systematically evaluating adversarial robustness using the typosquat/homoglyph stress mode released with this work.

Author Contributions: Conceptualization, G.F. and D.K.; methodology, G.F. and D.K.; software, G.F.; validation, G.F., D.K. and S.M.; formal analysis, G.F.; investigation, G.F.; data curation, G.F.; writing—original draft preparation, G.F.; writing—review and editing, D.K. and S.M.; visualization, G.F.; supervision, S.M.; project administration, S.M.; funding acquisition, S.M. All authors have read and agreed to the published version of the manuscript.

Funding: The research presented in this paper has been funded by the Research Committee (Special Account for Research Funds) of the Hellenic Open University through the project entitled “Enhanced URL Filtering with Machine Learning Techniques (MELEF)” (Grant Agreement No. 80702).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The benign domain list is from the public Tranco ranking and the malicious URLs from the public OpenPhish and URLhaus feeds (accessed 29 June 2026); these feeds update continuously. The experiment code, the exact `results.json` and parameter sweep, and the figure-generation scripts that reproduce all results in Section 4 are openly available on Zenodo at <https://doi.org/10.5281/zenodo.21027021>.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Antonakakis, M.; Perdisci, R.; Nadji, Y.; Vasiloglou, N.; Abu-Nimeh, S.; Lee, W.; Dagon, D. From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware. In Proceedings of the 21st USENIX Security Symposium, Bellevue, WA, USA, 8–10 August 2012; pp. 491–506.
2. Aljabri, M.; Altamimi, H.S.; Albelali, S.A.; Al-Harbi, M.; Alhuraib, H.T.; Alotaibi, N.K.; Alahmadi, A.A.; Alhaidari, F.; Mohammad, R.M.A.; Salah, K. Detecting Malicious URLs Using Machine Learning Techniques: Review and Research Directions. *IEEE Access* **2022**, *10*, 121395–121417. [\[CrossRef\]](#)
3. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 3982–3992.
4. Wang, W.; Wei, F.; Dong, L.; Bao, H.; Yang, N.; Zhou, M. MiniLM: Deep Self-Attention Distillation for Task-Agnostic Compression of Pre-Trained Transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2020; Volume 33.
5. Karapiperis, D.; Feretzakis, G.; Mitropoulos, S. PhishGraph: A Disk-Aware Approximate Nearest Neighbor Index for Billion-Scale Semantic URL Search. *Electronics* **2025**, *14*, 4331. [\[CrossRef\]](#)
6. Karapiperis, D.; Feretzakis, G.; Mitropoulos, S. PhishCluster: Real-Time, Density-Based Discovery of Malicious URL Campaigns from Semantic Embeddings. *Information* **2026**, *17*, 64. [\[CrossRef\]](#)
7. Karapiperis, D.; Feretzakis, G.; Mitropoulos, S. High-Throughput Semantic Embeddings for Real-Time Threat Campaigns. In Proceedings of the 17th International Conference on Information, Intelligence, Systems and Applications (IISA), Rhodes, Greece, 6–9 July 2026.
8. Maneriker, P.; Stokes, J.W.; Lazo, E.G.; Carutasu, D.; Tajaddodianfar, F.; Gururajan, A. URLTran: Improving Phishing URL Detection Using Transformers. In Proceedings of the MILCOM 2021—2021 IEEE Military Communications Conference, San Diego, CA, USA, 29 November–2 December 2021; pp. 197–204.
9. Wang, Y.; Zhu, W.; Xu, H.; Qin, Z.; Ren, K.; Ma, W. A Large-Scale Pretrained Deep Model for Phishing URL Detection. In Proceedings of the ICASSP 2023—2023 IEEE International Conference on Acoustics, Speech and Signal Processing, Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.
10. El Mahdaouy, A.; Lamsiyah, S.; Janati Idrissi, M.; Alami, H.; Yartaoui, Z.; Berrada, I. DomURLs_BERT: Pre-trained BERT-based Model for Malicious Domains and URLs Detection and Classification. *J. Netw. Syst. Manag.* **2026**, *34*, 36. [\[CrossRef\]](#)

11. Malkov, Y.A.; Yashunin, D.A. Efficient and Robust Approximate Nearest Neighbor Search Using Hierarchical Navigable Small World Graphs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2020**, *42*, 824–836. [[CrossRef](#)] [[PubMed](#)]
12. Subramanya, S.J.; Devvrit, F.; Kadekodi, R.; Simhadri, H.V.; Krishnaswamy, R. DiskANN: Fast Accurate Billion-Point Nearest Neighbor Search on a Single Node. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vancouver, BC, Canada; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2019; Volume 32.
13. Xu, Y.; Liang, H.; Li, J.; Xu, S.; Chen, Q.; Zhang, Q.; Li, C.; Yang, Z.; Yang, F.; Yang, Y.; et al. SPFresh: Incremental In-Place Update for Billion-Scale Vector Search. In Proceedings of the 29th Symposium on Operating Systems Principles (SOSP), Koblenz, Germany, 23–26 October 2023; pp. 545–561.
14. Jégou, H.; Douze, M.; Schmid, C. Product Quantization for Nearest Neighbor Search. *IEEE Trans. Pattern Anal. Mach. Intell.* **2011**, *33*, 117–128. [[CrossRef](#)] [[PubMed](#)]
15. Guo, R.; Sun, P.; Lindgren, E.; Geng, Q.; Simcha, D.; Chern, F.; Kumar, S. Accelerating Large-Scale Inference with Anisotropic Vector Quantization. In Proceedings of the 37th International Conference on Machine Learning (ICML), Virtually, 13–18 July 2020.
16. Wang, M.; Wang, Y.; Lv, L.; Yue, Q.; Xu, X.; Ni, J. An Efficient and Robust Framework for Approximate Nearest Neighbor Search with Attribute Constraint. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2023.
17. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, USA, 2–4 August 1996; pp. 226–231.
18. Campello, R.J.G.B.; Moulavi, D.; Sander, J. Density-Based Clustering Based on Hierarchical Density Estimates. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Gold Coast, Australia, 14–17 April 2013; pp. 160–172.
19. Cao, F.; Ester, M.; Qian, W.; Zhou, A. Density-Based Clustering over an Evolving Data Stream with Noise. In Proceedings of the 2006 SIAM International Conference on Data Mining, Bethesda, MD, USA, 20–22 April 2006; pp. 328–339.
20. Baer, A.; Finamore, A.; Casas, P.; Golab, L.; Mellia, M. Large-Scale Network Traffic Monitoring with DBStream, a System for Rolling Big Data Analysis. In *Proceedings of the IEEE International Conference on Big Data*; IEEE: New York, NY, USA, 2014.
21. Ribeiro, M.T.; Singh, S.; Guestrin, C. “Why Should I Trust You?”: Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), San Francisco, CA, USA, 13–17 August 2016; pp. 1135–1144.
22. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, Long Beach, CA, USA; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2017; Volume 30, pp. 4765–4774.
23. Capuano, N.; Fenza, G.; Loia, V.; Stanzione, C. Explainable Artificial Intelligence in Cybersecurity: A Survey. *IEEE Access* **2022**, *10*, 93575–93600. [[CrossRef](#)]
24. Alotaibi, S.R.; Alkahtani, H.K.; Aljebreen, M.; Alshuhail, A.; Saeed, M.K.; Ebad, S.A.; Almukadi, W.S.; Alotaibi, M. Explainable Artificial Intelligence in Web Phishing Classification on Secure IoT with Cloud-Based Cyber-Physical Systems. *Alex. Eng. J.* **2025**, *110*, 490–505. [[CrossRef](#)]
25. Karapiperis, D.; Feretzakis, G.; Kalles, D.; Mitropoulos, S. Explainable AI for URL Threat Detection in Healthcare Cybersecurity: A Case Study Using LIME and SHAP. *Stud. Health Technol. Inform.* **2025**, *328*, 246–250. [[CrossRef](#)] [[PubMed](#)]
26. Li, Y.; Huang, C.; Deng, S.; Lock, M.L.; Cao, T.; Oo, N.; Lim, H.B.; Hooi, B. KnowPhish: Large Language Models Meet Multimodal Knowledge Graphs for Enhancing Reference-Based Phishing Detection. In Proceedings of the 33rd USENIX Security Symposium, Philadelphia, PA, USA, 14–16 August 2024; pp. 793–810.
27. OpenPhish. Phishing Intelligence. Available online: <https://openphish.com/> (accessed on 29 June 2026).
28. URLhaus. Malware URL Exchange. Available online: <https://urlhaus.abuse.ch/> (accessed on 29 June 2026).
29. Le Pochat, V.; Van Goethem, T.; Tajalizadehkhoob, S.; Korczyński, M.; Joosen, W. Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In Proceedings of the 26th Annual Network and Distributed System Security Symposium (NDSS), San Diego, CA, USA, 24–27 February 2019.
30. Song, K.; Tan, X.; Qin, T.; Lu, J.; Liu, T.-Y. MPNet: Masked and Permuted Pre-training for Language Understanding. In *Advances in Neural Information Processing Systems (NeurIPS)*; Neural Information Processing Systems Foundation, Inc.: San Diego, CA, USA, 2020; Volume 33.
31. Feretzakis, G.; Karapiperis, D.; Mitropoulos, S. Real-Time Phishing Campaign Detection for Healthcare Organizations: An Explainable AI Approach Using Semantic Clustering. *Stud. Health Technol. Inform.* **2026**, *338*, 303–307. [[CrossRef](#)] [[PubMed](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.